APPLE II

微型计算机使用手册

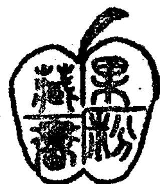
·陆煜钧·

·气象出版社·



APPLEⅡ微型计算机
使用手册

陆煜钧 编著





气象出版社

内容简介

本书重点介绍了在APPLE Ⅱ 微型计算机上用BASIC语言编制程序的方法及其在各方面的应用，特别在数据处理上的应用。全书主要通过例子帮助读者掌握APPLE Ⅱ 机的操作及应用，是使用APPLE Ⅱ机的一本实用工具书。

本书可供从事计算机及计算机应用的广大科技人员使用。



APPLE Ⅱ微型计算机使用手册

陆煜钧　编　著

责任编辑　黄丽荣

*　*　*

气象出版社 出版

（北京市白石桥路46号）

北京印刷一厂印刷

新华书店北京发行所发行　全国各地新华书店经售

*　*　*

开本：787×1092 1/32印张：12.75 字数：278 千字

1986年 3 月第 1 版　1986年 3 月第 1 次印刷

印数：1—42,000

统一书号：13194·0232　定价：3.10元

目　　录

前　言

APPLE Ⅱ微型计算机是当前流行的八位微型机之一。近年来，我国正在大量引进，同时也已引进、编译了若干使用手册，给开发应用带来了不少方便。但从目前接触到的手册来看，多数是从生产厂家角度出发介绍微机的一般使用方法，虽具有一定的广度，然其内容则过于庞杂。尤其是这一类使用手册，对于用微机进行数据处理方面的内容更是介绍得不够充分。本书就是想弥补这方面的不足，并从使用者的角度来编写APPLEⅡ 微机使用手册的。我们力求用最常用、最简单的BASIC语言进行编写，同时注意到内容安排合理、文字简明扼要，以帮助读者较快掌握 APPLE Ⅱ机的一般操作和应用，为今后进一步的开发应用打下一定的基础。此外，本书也可作为使用两个操作系统的一本实用工具书。本书仅一般地介绍了BASIC语言，重点放在APPLE Ⅱ机上所用语言的具体规定。另考虑到我国较多使用 APPLE Ⅱ机作数据处理的，故我们对数据处理方面的内容就介绍得详细些。特别在书中专用一章提供了一批数据处理程序，主要是常用的多元分析方面的程序；也适当介绍了几个有关人事及工资管理方面的程序，以供这方面工作的同志参考。为了便于交流，我们对计算机专业用语力求标准化，全书中的有关术语均以一九七七年人民邮电出版社出版的《英汉计算技术辞典》为准。

书中第四章第二节和第三节是刘功庆同志编写的，第四章的逐步回归程序和判别分析程序分别是王雅飞和李永顺同志提供的。全书承蒙国家气象局高级工程师吴贤纬同志校审，

并提出了不少宝贵的意见，作者谨致以衷心的感谢。

由于我们水平所限，谬误之处请读者批评指正。

作者

1984年12月

第一章　APPLE Ⅱ 微型机简介

§ 1.1　APPLE Ⅱ 微型机的构成及安装

APPLE Ⅱ 基本系统包括主机、磁盘驱动器（或磁带录音机）、键盘、显示器及打印机，它们分别用电缆与主机连接。

1.1.1　主机

主机是计算机的核心，其余都是外部设备。APPLE Ⅱ的主机与键盘是组装在一起的，内部通过电缆连接。键盘安装在主机最前面的位置上。打开主机的盖子就能看到内部结构，其中，后方最左边是开关型电源，输入交流电压110/220伏任选。电源开关就在机器后背，电源线是三芯线，功率消耗全载为60瓦，允许瞬间过载为79瓦。电源输出电压有四种：＋5伏，－5.2伏，＋11.8伏，－12伏。

主机底板的 8 个50芯连接插槽的前方，有一片特别大的集成电路块，这是 APPLE Ⅱ的主脑。它是美国MOS技术公司出品的6502微处理器，其基本指标如下：

基本指令：56条。

基本寻址方式：13种。

寻址范围：65536个字节（即64 K字节）。

基本字长：8 位。

运算速度：每秒50万次加、减法运算。

地址总线：16位平行。

数据总线：8 位并行、双向传送。

在中央处理单元(CPU)6502的正前方是6个只读存贮器(ROM)，共有12K存贮空间；其中2K是系统监控程序或APPLE自启动监控程序，其余10K是APPLE SOFT BASIC解释程序或APPLE II整型BASIC解释程序等。

在ROM的前面是APPLE II的随机存取存贮器(RAM)，总计有49,512个字节（48K字节）的存贮单元；只要在主机外接插槽上加插语言插件，就可把主机RAM扩充到64K。

APPLE II主板上还有其它一些元件和集成电路片，共同完成计算机之间信息交换、屏幕显示以及声响控制等功能。

主机板的最后面是8个50芯插槽，插槽编号从0到7,每个插槽可插入一块外部设备的插件(Card)用来扩展RAM和ROM,连接打印机或其它输入输出设备以及通讯设备等。

主机后方除电源插槽及开关外，还有连接视频显示的插孔以及用录音机输入、输出的插孔。

1.1.2 键盘

键数：52个键。为分清数字键零与字符键O,显示和打印时常常用0代表数字零。

使用码：大写的ASCII码（美国信息交换标准代码），见第二章附录（一）。

码数：91个。

转换键（SHIFT键）：2个。

特殊键：CTRL等6个键。

键盘和主电路板间以一条16芯扁平电缆相连接。键盘最前面有个红指示灯,指示主机电源是否接通，按下这个按钮，再打键盘上的字母，就是小写字母。

1.1.3 视频显示器

视频显示器可用彩色及黑白显示器，计算机专用的显示器，只要用信号电缆从主机后部的Video输出端，接到显示器的Video的输入端即可；若显示器用家庭电视机，则必须通过RF调制器相连接。

显示器有三种显示方式：

文本方式（TEXT mode)：24行，每行40个字符，共计960个字符，每个字符为5×7个点阵组成，具有白黑闪亮光标。

低分辨率图形方式：40行，48列，共有1920个方块，有16种颜色选择。

高分辨率图形方式：280行，192列，共计53760个点，可选择6种颜色。

1.1.4 磁盘驱动器

磁盘驱动器是微型机的重要外部设备。APPLE Ⅱ 机所用磁盘驱动器为DISK Ⅱ 系统的组成部分。磁盘驱动器通过扁平电缆连接到驱动器控制插件插脚上，每个驱动器控制板插件可以连接两个磁盘驱动器，插件上标有DRIVE 1的插脚接第一个驱动器，DRIVE 2 的插脚接第二个驱动器。驱动器电缆连到插件上以后，就把插件插到主机的外接插槽上去(注意：主机电源必须是断开的)，除零号插槽以外，其它插槽都可以插，一般插在第6号插槽上较好，APPLE机的许多软件都按这种选定编写。APPLE Ⅱ 主机可以插三块驱动器插件，亦即可以连接6个磁盘驱动器，但是一般情况下，连接2个磁盘驱动器就够了。本书介绍使用时均以2个驱动器为例，插件也插在6号插槽来考虑。DISK Ⅱ 所用磁盘（也称软盘）是5.25英寸大小的单面单密度磁盘，

可以贮存143K字节容量。

磁盘驱动器价格昂贵，结构精密，维修要较高技术和设备。磁盘驱动器主要由软盘控制器、步进马达、磁头和磁头的定位装置等组成，使用时应该注意：

(1) 轻拿轻放，避免剧烈震动，以免定位机构松动，特别是在需要运输时，一定要妥善包装；

(2) 驱动器工作时，不要移动；

(3) 不要在马达旋转过程中（即驱动器红灯亮时），插入或拿出软盘；

(4) 控制温升变化，温升变化范围应小于每小时15℃；

(5) 环境温度的要求一般在4—40℃之间；

(6) 特别要避免灰尘侵袭，不要把沾有灰尘的软盘片插入到磁盘驱动器中去，可用磁带清洗液或95%的酒精清洗磁头；

(7) 必须远离磁场发生源，要求离电视机至少60厘米以上；停止工作时，不要将软盘留在驱动器上。

1.1.5 打印机

打印机有各种类型，但都是通过扁平电缆连接到插件上，插件再插到主机的插槽上，一般插在第1号插槽（本书说明文字均已假定打印机插件已插在第1号插槽）。

各种类型打印机的使用方法不同，要看具体的使用手册。最便宜的打印机，如μ-80打印机只能打印字符，不能有效地打印汉字。流行的EPSON MX-80 Ⅱ型打印机是一种击打式点阵打印机，字符组成9×7点阵，除打印字符外，主机只要配上汉字库就可打印汉字；而且可以用程序控制打印的字符间距、行间距以及每行打印的字符数，亦能打印图形，又能顺行及逆行打印，速度较快。另一种型号的打印机EPSON

MX-100 则可打印宽行规格，每行最多可打印136个字符，打印速度更快一些。

以上介绍的是APPLE Ⅱ 机的基本系统。除此以外，APPLE Ⅱ 还有其它功能，一般是通过其它的插件连到有关设备上来实现，这将在下一节介绍。这里要强调的是，在安装机器时，特别是把插件插入外接插槽时，一定要先关闭电源，不然不但会损坏插件，甚至会损坏主机。插件脚不要用手摸，安装前最好用酒精擦干净，插入插槽要牢靠。

§1.2 APPLE Ⅱ微型机的主要功能及应用

1.2.1 功能及应用

(1) 利用上述APPLE Ⅱ 机的基本系统，可进行于数据和字符处理以及存取等操作，并作为一个独立的微型计算机系统使用。

(2) 可以在APPLE Ⅱ 的外接插槽插上Z-80插件，使APPLE Ⅱ 变成一个Z-80机，能运行CP/M操作系统，执行MICROSOFT BASIC语言，用于数据及字符处理。在APPLE Ⅱ 外接插槽插上MC 68000插件，APPLE Ⅱ机就可以作为准16位机使用，能运行MC 68000机的系统软件，也可作为MC 68000机的终端机。

(3) 在外接插槽上插上串行接口(EIA RS-232C)插件，可用于计算机通讯及外接串行输出设备；在半双工通讯下，每秒传送75—19200波特（Baud)。

(4) 在外接插槽上插上并行接口(IEEE-488标准总线接口) 插件，既用作与仪器仪表的通讯，又可用作与外部设备的接口。

(5) 在外接插槽上插上D/A，A/D转换控制板，主机

即可控制模/数和数/模转换，用于各种控制工作。

1.2.2 多种高级语言

在APPLEⅡ上可以运行多种高级语言，主要是：

(1) APPLE SOFT BASIC语言是浮点BASIC语言，这种语言简单易学，适合初学者使用；

(2) INTEGER BASIC语言是整型BASIC语言，运行速度比APPLE SOFT BASIC快，但用于数据处理有局限性，例如不能处理带有小数的数值等；

(3) MICROSOFT BASIC语言，是在APPLE机插入Z-80微处理器插件后可以运行的语言系统；

(4) APPLE FORTRAN语言，比美国标准的FORTRAN 77在某些方面更具有扩展性，适用于科学和工程计算；

(5) COBOL-80语言适宜用于处理数据繁多而运算简单的问题，主要功能是描述数据结构和分析处理大批量的数据，广泛应用于商业及文书表格处理；

(6) APPLE PASCAL语言，PASCAL语言集中体现了结构程序设计的原则，可用于数值计算，又适合描述系统软件的各种算法，常用来作为软件移植和教学的工具。

§1.3 APPLEⅡ微型机的发展

1.3.1 发展简史

1975年在美国加利福尼亚州旧金山市南部的卡帕提尼的小镇上，有年轻的两兄弟约伯·史蒂芬和奥斯尼雅克·史蒂芬，对当时刚刚兴起的微电脑非常感兴趣。他们利用业余时间组装了第一部微型机，受到了朋友们的热烈欢迎。同时，应朋友们的要求很快承接了再组装50部微型机的订单，并以30天为

期赊购了一批价值25000美元的零部件，开展了成批制造微型机的业务。为纪念约伯上大学时期曾在一家苹果园里做过工，他们将所研制的微型机命名为APPLE Ⅱ（即苹果Ⅱ型），随之建立了APPLE Ⅱ微型机公司；且将初创时期曾研制过的单板机也追认为APPLE Ⅰ 型机。所以，实际进入市场的微型机一开始就是APPLE Ⅱ。它于1977年进入世界市场，由于性能优良、价格便宜，很适合家庭、个人和小单位使用，被誉为微机发展的第一个里程碑。1979年，它在全美8位微机销售量中名列第一。1980年底，APPLE公司推出了APPLE Ⅲ。1981年秋天，美国联邦通讯委员会确认APPLE Ⅲ 的射频干扰超过标准而不准生产，故APPLE公司又研制了APPLE ⅢPLUS，这样APPLE Ⅱ的大部分软件也可以在它上面运行。1983年1月，APPLE公司推出APPLE Ⅱe 微型机，性能比APPLE Ⅱ又有较大改进，已可配置光标定位器。1984年初，APPLE公司又推出操作系统PRODOS，为APPLE Ⅱe 增加了硬盘机及窗口管理功能。1984年初，APPLE公司同时又宣布可以用16位微处理器65816或65802来改造旧的APPLE Ⅱ机。这两种微处理器都与6502微处理器完全兼容，特别是改造APPLE Ⅱ较为方便，只要拔下6502，插上65802，就可把原来的8位机变成16位机了。

微机的发展方向是组建计算机局部网络，实现计算机之间的数据通信以及共享昂贵的外设资源。1980年美国CORVUS公司开始研制OMNINET局部地区网络，1982年春天出售第一批正式产品。这样，APPLE Ⅱ 微机也能连到这个网络上，并组成APPLE Ⅱ与IBM PC机共存的局部网络。

1.3.2 各种类型

APPLE Ⅱ 机有几种不同的类型，它们有的是在电路板上做了改进，也有的是系统软件做了改进。例如旧式的APPLE Ⅱ 机就没有自启动ROM，而是APPLE系统监控程序ROM，这样，有些键盘功能（如ESC-I功能）就没法用；而APPLE Ⅱ PLUS型微机就具有自启动ROM，主机板上有APPLESOFT BASIC解释程序等；APPLE Ⅱe则还有自动引入诊断程序等功能。

目前国内用得较普遍的是APPLE Ⅱ PLUS型微机，本书介绍的内容以RAM为64K的APPLE Ⅱ PLUS微机为标准。

第二章 APPLESOFT BASIC语言

§ 2.1 APPLE Ⅱ上BASIC语言的几种工作方式

前面提到在APPLE Ⅱ PLUS型微型机的主印刷电路板的ROM中装有APPLESOFT BASIC的解释程序，当机器处于这种工作状态下时，在显示屏上出现的提示符是〕；机器还存在一种工作方式,即是处于监督程序方式,在显示屏上的提示符是＊；当用版本是3.3 的系统盘引入磁盘操作系统(DOS)时，会把INTEGER BASIC(整型BASIC)解释程序装入机器，当机器处于它的工作状态下时，显示屏上的提示符是〉。可以使这三种工作方式互相转换，在显示屏上表现为三种提示符的互相转换。

2.1.1 提示符〕与〉的转换

若机器提示符是〕，则键入INT命令,即转变为〉,表示转变为整型BASIC工作状态了；若机器提示符是〉；则键入FP命令,此时提示符变成〕,即机器已转入APPLESOFT BASIC工作状态。这两种提示符相互转换过程中,不破坏磁盘操作系统，但将丢失已打到机器中去的程序。

2.1.2 提示符＊与〕或〉的相互转换

若机器提示符是〕或〉，则键入CALL-151命令,变成＊提示符，其旁为闪亮的光标，表示机器已进入监控程序；若又要从＊返回到 APPLESOFT 工作状态，则可键入＊3D0G。这种提示符的相互转换方法，不破坏磁盘操作系

统，也不丢失已打到机器中去的程序。

若机器提示符是*，键入INT命令,则转为>；若机器提示符是*，键入FP命令，则转为]。这种转变方式，只保留了磁盘操作系统，已打入机器中的程序将全部丢失。

在提示符转换中，若键入FP命令后,显示的是"PROGRAM TOO LARGE"(程序太大)信息的话，则应该打入INT来使系统复位,然后再打入FP命令，即可顺利转换。

以下我们将介绍APPLESOFT BASIC的有关操作。

§ 2.2 磁盘操作系统

2.2.1 磁盘操作系统(DOS)的引入

磁盘操作系统控制了主机与磁盘机的信息交换，开机时首先碰到的就是把磁盘操作系统引导到主机存贮器内的问题。根据APPLE Ⅱ 机不同的型号,亦有多种不同的引导方法。有些机器很简单，如APPLE Ⅱ PLUS，在开机之前，先把随主机带来的系统盘（system master diskette）插入第一号驱动器，开机，这时DISK Ⅱ的红灯亮，有转动声，几秒钟后，在显示屏上会显示如下信息：

```
DOS VERSION 3.3                08/25/80

APPLE II PLUS OR ROMCARD    SYSTEM MASTER
(LOADING INTEGER INTO LANGUAGE CARD)
```

这表示DOS已进入存贮器。同时整型BASIC解释程序也已进入机器，即这时可以在机器上任意选用APPLE-

SOFT BASIC、整型BASIC或监督程序的工作方式了。

有时已经开机了，但由于某种需要，要求在某种工作方式状态下，重新引入DOS，有几种引导方式：

（1）若显示屏上出现的提示符为］或＞时，假定磁盘驱动器连接第6号插槽，则只要键入：

PR#6　再按RETURN键

或IN#6　再按RETURN键

就能启动磁盘驱动器，红灯点亮，即可将系统盘上的DOS引入机器。

（2）若显示屏提示符号*，则键入C600G，再按RETURN键，即会启动磁盘驱动器，红灯闪亮，表示DOS已引入。

（3）若提示符是*，则键入6 CTRL-P（指先键入6，再同时按CTRL键与P键），再按RETURN键，即可引入DOS。

2.2.2　磁盘操作系统的版本

DOS系统是美国APPLE公司专为APPLE Ⅱ微机使用而配置的操作系统，用6502的指令写成。APPLE DOS采用的是DOS 3版本，它经历了三个主要阶段：

（1）诞生阶段　APPLE Ⅱ公司于1978年6月和7月分别推出了DOS3.0和DOS3.1版本，以应付当时市场的急需。

（2）维护改进阶段　1979年4月和7月，APPLE Ⅱ公司先后推出了DOS3.2和DOS3.2.1版本，比DOS 3.1版本有所改进。

（3）更新阶段　1980年8月APPLE Ⅱ公司推出了DOS3.3版本，对DOS3.2.1版本进行了全面改进。它

将过去的13扇区改成16扇区，扩充了磁盘的使用容量，增加了存贮文件过程中自动进行校验的功能；为了能与DOS 3.2.1 版本的软件兼容，专门编制了13扇区到16扇区的转换软件，13扇区磁盘的引导程序等。

本书介绍的DOS系统指的是DOS3.3 版系统。

2.2.3 系统主盘及磁盘

前面已说过，为了引导DOS系统，必须先将系统主盘插入 1 号驱动器，然后开机引入。所以系统主盘对微机使用说来是关键磁盘，应该很好保存及使用，后面再介绍复制它的方法。另外，微机上常常要用到其它磁盘，它们也是密封在封套中的，目的是防尘，保护表面避免损伤，同时也防止盘片旋转时产生静电而使信息丢失，所以绝对不许把盘片取出封套。磁盘表面涂有磁性材料，记录信息时，磁性材料便是数据载体。磁盘外形如下图所示。

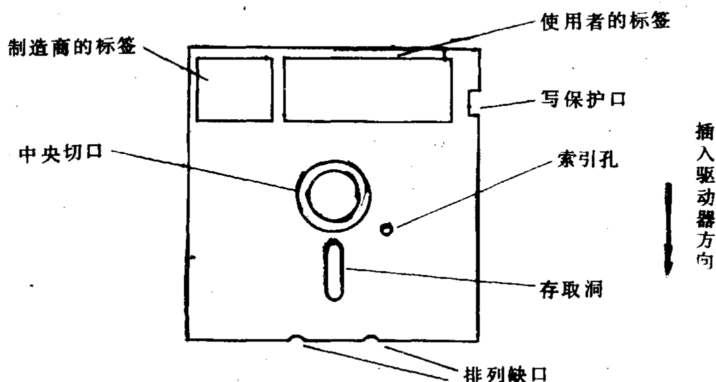


图 2.1

索引孔是进行磁盘分段时用来作为物理定位标志的，由于APPLE磁盘驱动器采用软分段，故索引孔在这里并没起作用。磁头读写窗口（存取洞）是磁头与磁盘接触读写信息的界面，是关键部位，绝对禁止用手接触窗口内磁盘的暴露部分，也应防止灰尘沾污。写保护切口是用来保护磁盘的被写入的，只要贴上胶纸，信息就写不到磁盘上去了，也就保护了磁盘上原有的信息。

使用磁盘的注意事项：

（1）磁盘的使用寿命为每一磁道可读/写 2×10^6次，平均工作时间为 300 小时；

（2）平时拿磁盘应该拿磁盘的边沿，不要加重压，不可弯曲，不用时及时插入纸外套中，立放在盒中；

（3）标签应写好之后再贴到磁盘上去，不要贴上标签后在磁盘上写字，以免损伤磁盘；

（4）软磁盘的保存温度最好在10—50℃，要避开磁场的干扰；

（5）已破损的磁盘不能再放到驱动器中使用，以免损坏驱动器的读写头；

（6）驱动器红灯亮时，绝对不能取出或放进磁盘。

§ 2.3 键盘的用法

在键盘板上除了英文字母键，数字键和算术运算键之外，还有一些专门用途的键：

红灯指示键及小写键

键盘板左下角那个开机时亮着红灯的键是表示主机电源已接通的指示灯，同时它也是个字母大小写的控制键，按下按键，转入小写状态，这时键入字母都是小写的了。

RESET键

这是复位键，在计算操作过程中要强迫中断某些操作时，可按CTRL-RESET键（指先按CTRL键不放，然后按RESET键；或同时按下两键也可）。

RETURN键

这是回车键，按了它，将对输入的字符及命令辨认，并存入内存，在每一条命令打入之后，都要按RETURN键。

SHIFT键

为功能变换键。在键盘上有的键可以有两种功能，按SHIFT键以后，再按那种键，就会把键上面所示那个字符键入主机。

CTRL键

是控制键，先按CTRL键不放，再按某些键，会起到不同的作用，如：

CTRL-X 把刚才打入主机的一行字符作废，这时虽然在显示屏上仍看到有那一行字符，但主机内存中已不存在了；

CTRL-G 主机发出“嘟”的一声；

CTRL-B 然后按RETURN，将从监督程序方式转到BASIC语言的操作；

CTRL-C 在程序列表的过程中途按这两个键，可使列表动作停止；

CTRL-S 在程序列表的过程中途按这两个键，可使列表动作暂时中止，再按空白键，列表动作又继续。

ESC键

ESC键是编辑键，它必须与别的键配合使用，但是不必同时按住，而是先按ESC键，放开后，再按别的键。这

个键主要用于屏幕编辑，例如：

ESC，I　即按了ESC键放开，再按I键，则屏幕上的光标向上移动一行；

ESC，J　按键方法与上述类似，但屏幕上光标向左移动一格；

ESC，K　屏幕上光标向右移动一格；

ESC，M　屏幕上光标向下移动一行。

上述屏幕编辑只要按ESC键一次，就进入编辑了，假如再按的不是I，J，K，M，REPT与SHIFT六个键中的任意一个键，就会脱离上述编辑方式了。要注意的是在以上编辑方式中只是控制光标的移动，并不改变屏幕及主机的内存。

ESC，E　把从光标所在的位置起到屏幕上该行的末尾之间的所有文字删除。

←键与→键

←键为向左移光标用，移过处的字符已从主机内存中删除，屏幕上则没有改变；

→键为向右移光标用，移过处的字符又重新进入主机内存。

REPT 键

是重复功能键，按住某字键，同时按REPT键，就会重复显示该字符；按REPT键，同时按→键或←键，就能快速移动光标，方便屏幕编辑。

程序编辑

用以上介绍的键盘功能就很容易对程序行进行编辑。方法是先调出这个程序行(即LIST(程序行))，然后按ESC键，和I键及J键，把光标调到这个程序行的行号之首，再

按→键，沿途对错误的字符修改，修改完后，按动→键（为加快移速，可同时按 REP T键），直到程序行的末尾，按RETURN键，即修改完毕。例如：

100PRINT“X=”；X

这个程序行中，打印语句写错了。为了改错，先键入LIST 100，把这个程序调出来，然后按ESC键，再按I键，使光标移到这个程序行位置，再按J键，把光标调到程序行之首1的位置，然后按→键移到M位置停住，按N键，再按→键，同时按REPT键，使光标快速移到程序行的末尾，按RET-URN键，即修改完毕。

假如要删去程序中某几个字符（设不在引号中的），则只要按上述调动光标，到需删除的那几个字符的位置处，按空格键就行。

若要在程序中插入字符，首先也是调出程序，按ESC键进入编辑状态，按I键及J键使光标移到程序之首，按→键，使光标位于待插入位置，按ESC键及I键把光标调到上一行位置，脱离编辑状态，键入要插入的字符，再按ESC键和J键到该行第一个位置，再按M键回到原来程序行位置，按→键，直到程序末尾，按回车键，完成了修改。例如：

100 PRINT　X

若是要在X之前插入“X=”；时，首先键入LIST 100 调出这个程序，然后按ESC键进入编辑状态，按I和J键，把光标调动到这行的行首，即1处，再按→键，把光标调到X处，再按ESC键，按I键，把光标向上抬一行，按任何别的键以脱离编辑状态，再按“X=”；和ESC键，再按J键，移到“位置，再按M键，回到原来的程序行位置，再按→键。若要快速移动，可同时按REPT键，直到程序末尾，按

RETURN键，结束修改。

§ 2.4 直接操作方式和命令

在直接操作方式下，不管系统处于何种工作方式，只要进入BASIC语言系统，以下的几种命令即可以立即执行：

2.4.1 PRINT

APPLE Ⅱ 的PRINT(打印）这个词，也可以简单地键入一个?,作用是一样的。

打印字符的方式是：

打入PRINT “字符串”,再按回车键(RETURN键)。

打印计算结果的方式是：

PRINT表达式（注意：不用引号),再按回车键。

在APPLESOFT方式下，每一行打印的长度不能超过255个字符，当输入长度接近255个字符时，会发出一连串“嘟”的声音，表示警告，当输入超过255个字符的时候，会印出“＼”的符号，并自动取消输入,就象按了CTRL－X键一样。

2.4.2 一行中用到几个命令

在一行中可以同时用到几个命令，例如，同一行中可以打印几组字符：

PRINT “A”:PRINT “AB”:PRINT “CDE”

按RETURN键后，即显示：

A

AB

CDE

2.4.3 执行没有行号的语句

可以键入没有行号的一组语句，直接得出结果。例如：

FOR I=1 TO 10：PRINT “A”;：NEXT：PRINT “PHEW!”

按RETURN键后显示：

AAAAAAAAAAPHEW!

直接执行方式除了一般的计算外，主要用于调试程序，检查程序执行过程中主机的内存。

2.4.4　NEW与HOME命令

NEW与HOME是完全不同的两种命令。NEW命令（按键NEW，然后按RETURN键），清除主机内存，但并不清除屏幕。NEW是常用的命令，特别在把程序输入主机之前，必须先键入NEW，以清除内存；键入HOME，然后按RETURN键，只是清除屏幕，使以后的显示更清楚些，并不清除主机内存，主机内存中仍旧保留了原先的内容。

2.4.5　CLEAR命令

它可把主机内存中的所有变量、数组和字符串变量清为零或空。CLEAR也可用于程序中。

2.4.6　RUN命令

是运行程序的命令。

2.4.7　PRINT　FRE（0）命令

FRE(0)用于释放内存单元函数，在直接命令方式下执行PRINT　FRE(0)是释放主机内存变量区的“废弃”单元，进行变量清理以得到可用的内存量(字节)，若可用RAM超过32767，则PRINT　FRE(0)得到负值，加上65536才是当时真正可用RAM；FRE(0)在程序方式中也可使用，用F＝FRE(0)语句达到直接命令方式的类似功用。

2.4.8 HTAB, VTAB命令

这是两个移动屏幕光标的命令：

HTAB n

执行这个命令就把显示光标在水平方向上移到n列位置，n的值在1与40之间。

VTAB n

执行这个命令把显示光标在垂直方向上移到n行位置，n的值在1与24之间。

HTAB与VTAB也可以用在程序方式中。

§2.5 程序操作方式（或称间接执行方式）和语句

程序操作方式指在APPLE Ⅱ上可以用APPLESOFT BASIC(简称APPLESOFT)语句，编造许多程序行，许多复杂的操作指令就包含在这些程序行内，程序行输入后并不立即执行，而必须打入运行(RUN)命令，才能得到结果，所以又称之为间接执行方式。

2.5.1 程序行和行号

编制程序必须有行号，它将决定程序执行的次序。行号必须在0与63999之间，写入时行号的先后没有关系，最终它们会自动按行号大小排列好；在一行中可以打入多个语句，语句之间用冒号隔开。

若打入一行语句之后，又要删除，则只要打入同样的行号，再按RETURN键即可；若打入同样行号的另一个语句，就把原先的语句顶替掉了；若要删除几个程序行，可以打入DEL命令，例如：

DEL 10, 50

再按回车键，就把10行到50行的语句都删掉了。程序行的修改请见§ 2.3 介绍。

2.5.2 数、字符、变量

数的表示

整数除符号外，有效位数5位，允许范围是-32767—32767；实数有效位为指数2位，底数9位，可用浮点数表示，允许范围是-1.7×10^{38}—1.7×10^{38}之间。

字符表示

字符须用双引号括起来，双引号括起来的字符序列称字符串，字符串长度为0—255 之间。

变量

变量名由大写字母和数字序列组成，第一个字符必须是大写字母。变量名最大长度可达238 个字符，但在APPLESOFT中只有最初2个字被认为有效。在变量名中不允许含有语言保留字，保留字见附录。变量名的最后一个字符代表变量的类型，字符$代表字符串变量，%表示整型变量，没有这两种符号就是实型变量。变量名相同，但是变量类型不同，仍表示是两个变量。例如A，A$，A%分别是实型变量，字符串变量和整型变量，它们可以在同一个程序中同时使用。

以下是变量名的合理和不合理的表示法：

合理			不合理		
A$	MN$	E6$	0$	M!$	77$
A%	B%	A1%	A$%	3I%	3D%
A	B	AA	0	7B	A#

2.5.3 表达式求值规则

表达式在求值时所按照的运算符（即从最高级的最内层

的括号计算开始，到最低级的布尔运算“OR”）执行优先次序如下：

（　）　括号
∧　乘幂
－　负号
*乘　　/除
＋加　　－减
＝等于　〈　〉不等于　〈　小于　〉大于
≤小于等于　≥大于等于
NOT　非
AND　且
OR　或

2.5.4　数组及DIM语句

在程序中使用数组时，前面必须用到数组说明语句DIM，例如：

10　DIM　A（9，11）

这是定义一个两维数组，每一维都是从零开始的，以上定义说明数组变量A在程序中用到的最大下标数目是A（9,11），可表达的变量为10×12＝120个。假如程序中某数组变量最大下标数不超过10，则APPLESOFT允许不要数组说明。

APPLESOFT允许最大使用88维的数组。

2.5.5　三种输入语句

输入语句主要有三种，将来介绍到磁盘文件时还将介绍从数据文件输入数据的语句。

赋值语句

也即LET语句，在APPLESOFT中的赋值语句可省去LET定义符。赋值语句的意思是指把表达式右边的数

值赋值给等号左边的变量，例如：

10 A＝10

20 V＝3.14159＊A∧2

又例如下面的赋值及打印语句：

```
10 X = 1
20  PRINT "X=";X
30 X = X + 1
40  PRINT "X=";X
50  END
```

运行的结果是：

X＝1

X＝2

键盘输入语句

即INPUT语句，在运行程序过程中，遇到INPUT语句处，主机将等待由键盘输入数据给变量，例如：

```
10   INPUT X
20   PRINT X * X
30   END
```

运行这个程序，结果是：

？9（若键入9）

81

再运行一次：

？4（若键入4）

16

在运行过程中，遇到程序中的INPUT语句，将显示一个问

号，即要求键入数据，这时从键盘输入数据，程序就往下执行了。

又例如：

```
10   INPUT X,Y,E
20   PRINT X,Y,E
30   END
```

运行这个程序，结果是：

? 1（若键入1）

??2（若键入2）

??3（若键入3）

1　　2　　3

这个例子说明INPUT语句后面可以要求输入几个变量值，但变量之间须要用逗号隔开。要注意从键盘打入的变量不要多于所要求输入的变量个数，并要求类型匹配和输入次序正确。

READ语句，DATA语句，RESTORE语句

READ语句是读数语句，在程序中必须与置数语句DATA配合使用，依次将程序中DATA语句所提供的数据序列赋值给READ语句中的各个变量。例如：

```
10   READ R1,R2,R3,R4,V
20   DATA 100,150,240,360,24
30 I = (1 / R1 + 1 / R2 + 1 / R3 +
      1 / R4) * V
40   PRINT "I=";I
50   END
```

运行这个程序的结果:

I＝0.566669

一个程序中，DATA语句中的数据可以比READ语句中的变量多，但是不能少，否则运行就会出错，还要注意变量类型要与数据序列类型一致，互相匹配。

RESTORE语句是可以恢复使用DATA语句的一种语句，例如:

```
10  READ A,B,C,D
20  RESTORE
30  READ E,F
40  PRINT A,B,C
50  PRINT D,E,F
60  DATA 1,3,5,7
70  END
```

运行这个程序结果:

1	3	5
7	1	3

即第10行读语句读完第60行的数据，第20行的RESTORE又复位到这批数据的开始处，变量E，F又可从第一个数据开始读数据。

三种输入语句的比较

三种输入语句各有各的用处，在有的程序中几乎同时用到这三种输入语句。三种输入语句的特点是:

LET语句适用于给少量数据赋值，在有大批数据的情况下就显得繁琐，输入效率不高。

READ/DATA 语句适用于成批数据的输入，在屏幕

上可对所用数据成批校对。

INPUT语句使用灵活性大，在程序执行过程中可以灵活输入参数，边试算边修改，缺点是人工键盘输入，运行速度慢。

2.5.6 控制语句

GOTO（无条件转向）语句

格式是：

GOTO语句〈行号〉

程序中遇到GOTO语句，就改变了程序执行的方向，可能转到前面的程序行去执行程序，也可能跳过几个程序行，执行后面的程序，例如：

```
10   INPUT X
20   PRINT X
30   GOTO 10
40   END
```

这个程序所起作用是只要键盘不断输入数据，就会不断打印出来，永无止息。

条件语句

有几种格式：

① IF（表达式）THEN语句：语句：语句

以上语句中的表达式可以是算术表达式、字符串表达式或条件表达式。若表达式的值不为零时，执行THEN后面的语句和它后面并列的语句(可以有许多个)，若表达式的值为零时整个语句就不执行了。

表达式为零是指下列三种情况之一：算术表达式计算出

的绝对值小于2.93873E－39；字符串表达式的结果为空串时；条件表达式的条件不满足。

IF　THEN 型条件语句使用例：

```
10   INPUT "A=";A
20   IF A < 8 THEN B = 4: PRINT B:
       GOTO 50
30 B = 16
40   PRINT "B=";B
50   END
```

运行这个程序的结果：

A＝6（假如键盘输入6）

B＝4

再运行这个程序：

A＝10（假如键盘输入10）

B＝16

② IF（表达式）THEN（语句行号）

表达式成立，则程序转向到指定的行号执行。

③ IF（表达式）GOTO（语句行号）

表达式成立，则程序转向到指定的行号执行。

GOSUB语句

格式是：GOSUB（语句行号）

在一个程序中，有时常常要重复使用一段程序，为了简化使用，把这种被重复使用的一段程序编成子程序形式，在主程序要用到它时就调用它，调用完后，继续往下执行主程序。调用时的语句即为GOSUB(语句行号)，子程序结尾是RETURN语句。

例如：求任意三个数的阶乘的和：

$$S = A! + B! + C!$$

```
10  GOSUB 100
20 A = P
30  GOSUB 100
40 B = P
50  GOSUB 100
60 C = P
70 S = A + B + C
80  PRINT "=";S
90  END
100 P = 1
110  READ M
120  FOR J = 1 TO M
130 P = P * J
140  NEXT J
150  PRINT "+";M;"!";
160  RETURN
200  DATA 2,3,4
```

程序中从100行到160行就是相对独立于主程序的子程序，所起的作用是读取一个数据，求它的阶乘。第10行调用这个子程序，算得2!后，返回执行第20行，把2!值赋给变量A，继续往下执行。运行这个程序的结果是：

$$+2! + 3! + 4! = 32$$

子程序可以嵌套子程序，嵌套深度最多可达24层。对某子程序说来，根据需要可以有不同的入口。

计算转向语句

格式：ON(变量)GOTO(语句行号1)，(语句行号2)，(语句行号3)，…

这种计算转向语句是按算术表达式中变量的值选择转向行号，当变量为1时，转向行号1去执行；变量为2时，转向行号2去执行；变量为3时，转向行号3去执行，假如变量为零或程序中没有指定的行号时，就执行下一个语句。例如：

```
10 B% = 4
20  PRINT B%
30 A% = B% - 2
40  ON A% GOTO 10,70,150
70  PRINT B%
80 B% = 5
90  GOTO 30
150  PRINT B%
160 B% = 3
170  GOTO 20
```

运行这个程序结果是：

4
4
5
3
4
⋮

计算转子程序语句

格式：ON(变量)GOSUB(行号1)，(行号2)，…

这种计算转子程序语句是按算术表达式中变量值选择调用子程序行号，作用与计算转向语句类似，与计算转向语句统称开关语句。

自动纠错语句

格式：ON ERR GOTO (行号)

程序中有了自动纠错语句后，当程序运行中遇到错误时，会自动转移到预先设计好的纠错程序中去，纠错程序末尾是RESUME语句，使得可以回到出错的语句处，以便继续执行程序，也可以使用GOTO语句，将程序转向到指定的行号继续执行。例如：

```
ULIST

10  ONERR  GOTO 100
20 A = 30
30 B = 0
40 C = A / B
50  PRINT A: PRINT B: PRINT C
60  END
100 B = 20: RESUME
```

运行这个程序的结果是：

30

20

1.5

循环语句

循环语句用于多次重复执行同一指令或同一组指令，称为程序循环，其一般形式如下：

10 FOR I=A TO B STEP C

20
30 } 循环体
⋮

```
100 NEXT I
```

因为它是从FOR开始，以NEXT结尾，亦称FOR-NEXT循环。其中 I 是循环控制变量；A是循环变量初值；B是循环变量终值；C是循环变量步长。第10行统称循环说明语句。第100行的NEXT是循环终端语句。在循环说明语句与循环终端语句之间的程序行称为这个循环结构的循环体。

循环结构的注意之点：

① 循环说明语句中的A，B，C值可以是个数值，也可以是变量或表达式，但是要求在执行循环之前已被赋值。C值为1时可以省略STEP；

② FOR与NEXT必须成对出现，NEXT后面的循环变量可以不写；

③ 循环语句最重要的功能是指定循环体执行的次数。

$$\text{循环次数} = \frac{B - A}{C} + 1$$

例如：

```
10  FOR I=1 TO 10
20  PRINT
30  NEXT I
```

运行这程序的结果是打了10个空行。

又例如：

```
10  FOR I=1 TO 10
20  PRINT I
30  NEXT I
```

运行这个程序的结果是打印出循环变量的值：

```
1
2
```

3

4

5

6

7

8

9

10

又例如:

```
10  FOR I=1 TO 5
20  PRINT "A"
30  NEXT I
```

运行这个程序的结果是打印 5 个字符A:

A

A

A

A

A

④ 循环可以嵌套,实现多重循环,但嵌套不能交叉,嵌套不得超过10层。有了多重循环可以简洁地处理许多复杂的问题。

例如:计算并打印乘法九九表:

```
10  FOR A = 1 TO 9
20  FOR B = 1 TO 9
30 P = A * B
40  PRINT A;"*";B;"=";P
50  NEXT B
```

```
60   NEXT A
70   END
```

又例如：若有任意个数，要求按大小排队。

```
10   INPUT "N=";N
20   DIM X(N)
30   FOR I = 1 TO N
40   READ X(I)
50   NEXT I
60   PRINT "I","X"
70   FOR I = 1 TO N
80   FOR J = I TO N
90   IF X(I) >  = X(J) THEN 110
100  T = X(I):X(I) = X(J):X(J) = T

110  NEXT J
120  PRINT I,X(I)
130  NEXT I
140  END
150  DATA 240,303,56,14,23,6
160  DATA -7,-4,4,2
```

运行这个程序的结果是：

```
N = 10
I       X
1       303
2       240
3        56
4        23
5        14
```

6	6
7	4
8	2
9	－4
10	－7

当然，DATA语句应该提供所需要的数据。

END语句与STOP语句

END语句一般放在程序末尾，表示程序运行到此结束，也可以不写，程序执行到末尾自然终止；当然END语句也可能出现在程序中间，在一定条件下，程序执行到此就结束了。

STOP语句出现于程序中间，是为调试程序的需要而设置的断点，程序执行到此暂时中止，以便检查这时各个变量的值，判断程序的执行是否正确，要继续执行程序时只要键入CONT命令就可以了。程序执行到STOP语句时，机器发出“嘟”的一声，以示提醒，并显示BREAK IN(程序停止执行所在的语句行号)，表示程序在哪一行中断的。

要注意的是，一般在程序末尾可以不加END语句，但是若末尾后面还有子程序时，则在子程序之前必须要加END语句，这在用到那些已编成子程序形式的程序库（如本书第四章），写主程序时一定要注意末尾写END语句。

2.5.7 REM语句

是注释语句，程序执行中并不执行这类语句，这类语句的作用是向使用者说明程序的功能或限制等，以便更方便地阅读程序。

2.5.8 PRINT语句

是打印输出语句，把机器内存的数据（data）输出到屏

幕、打印机或其它外部设备上。打印语句要注意与格式配合使用：

按标准间隔位置输出

打印输出变量之间以逗号分隔时，则按APPLE 机预先设定的标准位置三列输出，第一个位置是第1—16列，第二个位置是第17—32列，第三个位置是第33—40例。而且规定，第二个位置只有在第16列为空时才发生作用，第三个位置只有在第24—32列为空时才发生作用。例如：

```
10 A$ = "THREE"
20 B$ = "VISIONARY"
30 C$ = "MICE"
40  FOR I = 1 TO 4
50  PRINT A$,B$,C$
60  NEXT I
70  PRINT "GOOD!"
80  END
```

运行的结果是：

```
THREE           VISIONARY
MICE
THREE           VISIONARY
MICE
THREE           VISIONARY
MICE
THREE           VISIONARY
MICE
GOOD!
```

其中每一行第一个字符串及第二个字符串都打在标准位置，

由于第二个字符串太长，影响到第三个字符串不能打在标准位置上，而只能换行了。

按紧凑格式输出

打印输出变量之间用分号分隔时，则按紧凑格式输出，输出结果之间不留空白。例如：

```
10 A = 5
20  PRINT A;56;"ABC"; SIN (A)
30  END
```

运行结果是：

556ABC -.958924274

PRINT语句末尾的标点符号的作用

① 在最后一项不置标点符号时，即自动换行输出。例如：

```
10  PRINT 1, 3
20  PRINT 2*2, 5*5
30  END
```

运行的结果是：

```
1        3
4        25
```

② 在最后一项后面用分号时，则不换行，下列的输出紧接前行输出。例如：

```
10  PRINT "Y=";
20  PRINT 2;"*X1";"+";
30  PRINT 4;"*X2"
40  END
```

运行结果是：

Y＝2＊X1＋4＊X2

③ 在最后项后面是逗号时，打印完后不换行，在标准位置打印。例如：

```
10   PRINT "X", "Y",
20   PRINT "X+ Y"
30   END
```

运行结果是：

```
     X          Y          X+ Y
```

④ 打空行

```
10   PRINT
```

运行结果是打一个空行。

按指定位置输出

① SPC函数

是一个具有跳过空白位置功能的函数，必须用在PRINT之后。例如：

```
10   PRINT  SPC(15);  "TOP"
```

表示在一行先空15格，然后打印TOP字符串。SPC后面括号中可以是数值、变量或算术表达式，需打印的字符与SPC函数之间必须用分号隔开。

② TAB函数

是具有排列打印格式功能的函数，它规定了打印从哪里开始。TAB函数用在PRINT语句之后，若使用多个TAB，则它们之间也用分号隔开，TAB后面括号中可以是数值、变量或算术表达式。例如：

```
10   FOR I = 1 TO 5
20   PRINT  TAB( 10);I; TAB( 20);I
```

```
      + 1; TAB( 30);I + 2; TAB( 4
     0);I + 3
30  NEXT I
40  END
```

运行结果是:

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7
5	6	7	8

TAB函数括号中数值超过40就不起作用，所以当表格宽度超过40列时，可用 SPC 函数规定打印格式。

2.5.9 DEF 语句

这个语句是在程序中定义一个新函数。

格式是:

DEF FN变量名(X)=算术表达式

其中新函数的名字是FN变量名(X)，X是形式参数，它必须是实型变量。在程序的前面有了这个新函数的定义后，以后就可调用这个新函数了。例如:

```
10  DEF  FN P(X) =  INT (100 * X +
     0.5) / 100
20 R = 20
30 A = 2 * 3.14159 * R:B = 3.1415
     9 * R * R
40  PRINT  FN P(A): PRINT  FN P(B
     )
```

程序运行的结果是输出两位小数（经过四舍五入）的值：

125.66

1256.64

要注意只有用DEF语句定义过的函数才能以FN变量名（X）的形式出现在表达式中。

2.5.10 CLEAR 语句

是清除变量语句，它将把内存中所有变量、数组和字符串变量清为零。

2.5.11 INVERSE，FLASH和NORMAL

为了在屏幕显示时达到醒目作用而专门提供的功能。它们可以在程序中作为语句使用，也可以不在程序中而独立地作为命令使用。

INVERSE 用了这个命令或语句，可使屏幕上任何由PRINT命令或语句所显示的资料变成白底黑字；但是键盘输入的资料则还保持正常显示。

FLASH 用了这个命令或语句，可使屏幕上任何由PRINT命令或语句所显示的资料不住地闪亮；但是键盘输入的资料则还保持正常显示。

NORMAL 可把以上两种显示格式恢复为正常的显示格式。

2.5.12 PEEK和POKE

PEEK的功能是可以读到APPLE Ⅱ 内存中任何位置的值，例如PRINT PEEK(222）把内存222处的值打印输出；POKE语句则可把数值（必须在0—255之间）放入RAM中去，例如POKE 8000，A是把变量A的值放到内存8000处。被写入到内存的值，可以是一个数字或变数或算术表达式可产生的值。数值范围要在0—255之间。

§ 2.6 APPLESOFT的函数

APPLESOFT 语言中允许使用的函数主要有两大类，即算术运算函数和字符串运算函数。

2.6.1 算术运算函数

ABS(X)

得到X的绝对值

EXP(X) 指数函数

得到e^x值，其中e 取值为2.78289(即自然对数底)

LOG(X) 自然对数函数(即LN(X))

INT(X) 取整函数

得到小于或等于X值的最大整数。

例如： INT(3.56)=3

INT(-3.56)=-4

若要求对一个数作四舍五入处理，可以用INT(X+0.5)来做到。例如：

INT(3.14159*100+0.5)/100=3.14

对于任意数值要求按四舍五入处理指定小数位（例如两位）的严格表达式是：SGN(X)*INT(ABS(X)*100+0.5)/100

RND(X) 随机函数

得到大于零与小于1之间的实型随机数。

SGN(X) 符号函数

决定一个数的符号，即：

$$SGN(X)=\begin{cases} -1 & \text{当}X<0\text{时} \\ 0 & \text{当}X=0\text{时} \\ 1 & \text{当}X>0\text{时} \end{cases}$$

SQR(X)

得到X的平方根（X≥0）

三角函数(即SIN(X)，COS(X)，TAN(X)，ATN(X))

X的单位是弧度。

2.6.2 字符串运算函数

ASC(A$)

得到A$中第一个字符所对应的ASCII码。

例如：

ASC("ABC")=65

CHR$(X)

得到X值所对应的ASCII字符串。X的值必须在0—255之间，X≥128时，所产生的ASCII字符串为X值按128取模。

例如：

CHR$(65)= A

CHR$(193)= A

LEN(A$)

得到字符串A$的长度，包括空白间隔在内。

例如：

LEN("ABCD")=4

LEFT$(A$，X)

得到A$中最左边的X个字符，A$及X的值均只能在1—255之间。

例如：

LEFT$("ABCD",2)= AB

RIGHT$(A$，X)

得到A$中最右边X个字符，A$及X的值均只能在1—

255之间。

例如：

RIGHT$（“ABCD”,3）=BCD

MID$(A$，X1，X2)

得到A$中第X1个字符开始的向右连续X2个字符组成的字符串。

例如：

MID$（“ABCDEF”,3，2）=CD

STR$(X)

将数值X转换成相应的字符串。

例如：

STR$（41）=41　（等号右边的41是字符串了）

STR$（41000000000）=4.1E+10等号右边的也已是字符串。

VAL(A$)

将A$转换成对应的数值。

例如：

VAL（“-0.041”）=-0.041

VAL（“4.1E4 BASE”）=41000

VAL（“BASE4.1E4”）=0

字符串之间用“十”符号表示字符串的连接

例如：

```
10 A$ = "SHANG"
20 B$ = "HAI"
30 C$ = A$ + B$
40  PRINT C$
50  END
```

运行的结果是输出：

SHANGHAI

§ 2.7 其它的常用函数

数值运算中还经常用到另外一些函数，对这些函数APPLESOFT并不直接提供，但是可以用APPLESOFT已提供的上述基本函数很方便地组合起来得到。若是在程序中用DEF语句定义这类组合而成的函数，使用起来就更方便了。下面列举这类常用组合函数的表达式：

ARCCOS(X)=－ATN(X/SQR(－X*X+1))+1.5707633计算X的反余弦函数值，必须ABS (X)<1

ARCCOT(X)=－ATN(X)+1.5707633 计算X的反正切函数值。

ARCCOSH(X)=LOG(X+SQR(X*X－1)) 计算X的反双曲余弦函数值（ABS (X)≥1）。

ARCCOTH(X)=LOG((X+1)/(X－1))/2 计算X的反双曲余切函数值(ABS (X)>1)。

ARCCSC(X)=ATN(1/SQR(X*X－1))+(SGN(X)－1)*1.5707633 计算X的反余割函数值（ABS (X)>1）。

ARCCSCH(X)=LOG((SGN(X)*SQR(X*X+1)+1)/X)计算X的反双曲余割函数值（X>0）。

ARCSEC(X)=ATN(SQR(X*X－1))+(SGN(X)－1)*1.5707633计算反正割函数值(ABS (X)≥1)。

ARCSECH(X)=LOG((SQR(－X*X+1)+1)/X) 计算X的反双曲正割函数值(0<X≤1)。

ARCSIN(X)=ATN(X/SQR(－X* X+1))

计算X的反正弦函数值(ABS(X)<1)。

ARCSINH(X)=LOG(X+SQR(X*X+1))计算X的反双曲正弦函数值。

ARCTANH(X)=LOG((1+X)/(1-X))/2计算X的反双曲正切函数值(ABS(X)<1)。

COSH(X)=(EXP(X)+EXP(-X))/2 计算X的双曲余弦函数值。

COT(X)=1/TAN(X) 计算X的余切函数值（X<>0)

COTH(X)=EXP(-X)/(EXP(X)-EXP(-X))*2+1 计算X的双曲余切函数值（X<>0)。

CSC(X)=1/SIN(X) 计算X的余割函数值(X<>0)

CSCH(X)=2/(EXP(X)-EXP(-X))计算X的双曲余割函数值（X<>0)。

$LOG_a(X)$=LOG(X)/LOG(a) 计算X的以a为底的对数值（a>0，X>0)。

$LOG_{10}(X)$=LOG(X)/2.30258509 计算X的常用对数值（以10为底)，必须X>0。

$MOD_a(X)$=INT((X/a-INT(X/a)*a+0.05)*SGN(X/a) 计算X被a除之后的余数(a<>0)，即取模，上例称为XMODa。

SEC(X)=1/cos(X) 计算X的正割函数值（X<>π/2)

SECH(X)=2/(EXP(X)+EXP(-X)) 计算X的双曲正割函数值。

SINH(X)=(EXP(X)-EXP(-X))/2 计算X的

双曲正弦函数值。

TANH(X)=-EXP(-X)/(EXP(X)+EXP(-X))*2+1　计算X的双曲正切函数值。

§ 2.8　程序编制及调试

在§2.3介绍键盘使用时已简单地介绍了程序的编制，这一节作较详细介绍。

LIST 命令

LIST命令可把主机内存中的程序显示或打印出来，LIST命令有如下多种形式：

① LIST　（然后回车，以下相同）

把内存中全部程序行都列印出来。

② LIST　行号

把内存中某一行号的程序语句列印出来。

③ LIST　行号1，行号2

把内存中从行号1到行号2的某段程序行都列出来。

④ LIST　行号1，

把内存中从行号1开始到程序结束的全部程序行都列出来。

⑤ LIST，行号2

把内存中行号2以上的所有程序行全部都列印出来。

在列印程序行的过程中若要求暂停列表可按CTRL-S键，再按其它任何键，又会继续列表；若是按CTRL-C键，则会强迫中断列表。

空格

空格在APPLESOFT的保留字和表达式中，是没有任何意义的，故在程序编辑中可不考虑空格的位置。但假如

是在行号中的字符，编辑时就必须注意，空格也是字符。在APPLE Ⅱ 屏幕上显示的程序行两边都留有空位，假如程序编辑时，在程序行两边正好是引号内的空位，则在编辑时需注意空格的安排。方便的做法是在编辑前，先用命令POKE 33，33设定屏幕宽为33列，使程序两边不留空位，这样就不会出现上述困难了。编辑结束，打入命令POKE33，40恢复到正常宽度。程序中字符增、删、改的方法前面已介绍过了，不再重复。

程序调试

为了检查程序编制是否正确，可通过运行动态检查程序逻辑是否有错。检查错误的方法是多种多样的：可以在程序中安排STOP 语句，执行到一定阶段暂停运行，可以调出机器内存检查计算的中间结果等。还有一种常用的方法是跟踪调试，跟踪整个程序的运行过程。跟踪命令为TRACE。例如：

先键入命令TRACE，使主机预置跟踪状态，然后打入程序如下：

```
10   FOR I = 1 TO 3
20   FOR J = 1 TO 4
30   FOR K = 1 TO 3
40   READ A(I,J,K)
50   PRINT A(I,J,K)
60   NEXT
70   NEXT
80   NEXT
90   DATA 1,2,3,4,5,6,7,8,9,10,11,
      12,13,14,15,16,17,18,19,20
100   DATA 21,22,23,24,25,26,27,28
      ,29,30,31,32,33,34,35,36
```

运行情况如下：

```
#10 #20 #30 #40 #50 1
#60 #40 #50 2
#60 #40 #50 3
#60 #70 #30 #40 #50 4
#60 #40 #50 5
#60 #40 #50 6
#60 #70 #30 #40 #50 7
#60 #40 #50 8
#60 #40 #50 9
#60 #70 #30 #40 #50 10
#60 #40 #50 11
#60 #40 #50 12
#60 #70 #80 #20 #30 #40 #50 13
#60 #40 #50 14
#60 #40 #50 15
#60 #70 #30 #40 #50 16
#60 #40 #50 17
#60 #40 #50 18
#60 #70 #30 #40 #50 19
#60 #40 #50 20
#60 #40 #50 21
#60 #70 #30 #40 #50 22
#60 #40 #50 23
#60 #40 #50 24
#60 #70 #80 #20 #30 #40 #50 25
#60 #40 #50 26
#60 #40 #50 27
#60 #70 #30 #40 #50 28
#60 #40 #50 29
#60 #40 #50 30
#60 #70 #30 #40 #50 31
#60 #40 #50 32
#60 #40 #50 33
#60 #70 #30 #40 #50 34
#60 #40 #50 35
#60 #40 #50 36
#60 #70 #80 #90 #100
```

退出跟踪的命令是NO TRACE，即取消跟踪，恢复正常，使 DOS 命令能恢复作用。

列表打印

要把内存中的程序输出到打印机上的步骤是：

① 打开打印机电源和接通打印状态。

② 键入PR＃1，联通打印机。

③ 若在打印机上是40列输出，则只要键入LIST 就行了；若要求80列输出，则应该先键入命令POKE 1656＋SLOT，80其中SLOT 指的是打印机插件插的槽号，一般插在1号插槽上，所以一般打印80列的命令是 POKE 1657，80，回车，然后键入LIST就可，这时程序只在打印机输出，屏幕上没有程序显示。若要恢复40列打印输出和屏幕可见输出，则只要键入POKE 1657，40（假定打印机插件是在1号插槽）即可。

§ 2.9 数据处理型的程序编制技巧

2.9.1 程序编制方法

计算机的运算过程与人们算题的思路类似，即：

(1) 首先要输入原始数据。所以在程序的第一部分要有输入数据语句，输入数据的语句多种多样，根据需要选择，使用数组的，必须要有数组说明语句。另外要指出的是，APPLESOFT语言编制的程序在运行时，首先自动把各个变量取零，所以程序中变量常不必先赋值零。

(2) 要有解题的数学模型，这常是程序编制中的核心问题，要用计算机解题，就要设计好适合于计算机计算的算法模型。

(3) 要设计好计算结果的输出。为了要使计算结果清楚、

明了、正确地打印输出，要在程序中很好地安排打印格式语句。

2.9.2 程序编制技巧

一般认为，好的程序应该是运行速度较快，所占内存较小；结构清楚，通用性强，维护修改方便等，以下简单介绍数据处理类型的程序设计中应该注意的几点：

(1) 程序中尽量避免使用常数，特别是整常数，而应该及早把这类常数赋值给变量，以后再用这个值时就可以用变量代替了，这样可以加快运算速度，有时还节省内存。

(2) 尽量使用在程序最前面已经用到的变量，可以加快运算速度。

(3) 程序中最常用到的子程序按排的行号应该最小，能加快调用的速度。

(4) 在NEXT语句后面省去循环变量，可以提高运算速度。

(5) 能在循环语句外面执行的语句切勿移到循环内部去执行，可以加快运算速度。

(6) 注释语句中的说明要尽量简洁，以节省内存。

(7) 数组中尽量用到第零号元素；变量尽可能重复使用，以节省内存。

(8) 每个程序行号要占用5个字节的内存，因此应尽量节省使用程序行，在某些情况下可以把几个语句放在一个程序行内，用冒号隔开。缺点是编辑、修改和阅读较困难。

(9) 采用READ/DATA语句的输入方式占用内存较多，可选用INPUT语句，或从数据文件（见后面介绍）输入数据。

(10) 在程序中尽可能用开关语句(ON GOTO, ON

GOSUB)，可以加快转移运算速度。

§ 2.10 APPLESOFT出错信息表

由于BASIC解释程序对BASIC语言的程序是执行一句，检查一句，因此对程度执行过程中，所遇到的语法错误或运算过程错误，系统将发出警告，并显示错误信息，给出出错类型和相应的语句行号，并停止程序的继续执行。系统给出的错误信息，按整数BASIC和APPLESOFT及DOS三类输出。这里只将APPLESOFT出错信息含义列举如下。

(1) ? BAD SUBSCRIPT ERROR数组变量定义不正确。常见于程序中用到的数组变量元素超出了已说明的范围（即所用下标变量的下标超出语句所定义数组的范围）例如：

```
10 DIM X (13)
20 FOR I=1 TO 20
30 READ X (I)
40 NEXT I
50 DATA 13, 20, 11, 2,4,5,3,7,4,6
60 DATA -5,6,77,8,11,3,3,-4,0,-5
```

运行这个程序就会出错。

(2) ? CAN'T CONTINUE ERROR 若根本没有程序，或已经出现了严重错误，或者已经修改了程序之后，还用CONT命令执行程序时就会出错，称为不能继续运行错。

(3) ? DIVISION BY ZERO ERROR 分母为零错，因为在任何数值计算都禁用零除。

(4) ? FORMULA TOO COMPLEX ERROR 公式太复杂错。一个语句中若是IF (字符串)THEN型方式已执行了两次以上，则出错。

(5) ? ILLEGAL DIRECT ERROR非法命令错。把INPUT,DEF FN或GET作为直接操作方式的命令使用就出错。

(6) ? ILLEGAL QUANTITY ERROR非法数值错。有如下一些情况：

①数组的下标用了负值；

②对零或负值进行LOG函数的运算；

③对负值进行SQR函数运算；

④A ↑ B中A为负数，B不是一个整数；

⑤ 在MID$,LEFT$,RIGHI$,WAIT,PEEK,POKE，TAB和SPC等函数中使用了不恰当的参数。

(7) ? NEXT WITHOUT FOR ERROR指在NEXT语句前面没有FOR语句与之对应。

(8) ? OUT OF DATA ERROR数据不够用。在执行READ语句时，DATA语句中的数据已经用完了。

(9) ? OUT OF MEMORY ERROR超出内存错。原因可能是：

① 程序太大，变量太多，字符串运算中没有适当地用A＝FRE(0)函数清除作废的单元；

② FOR循环嵌套超过10层，调用子程序嵌套超过24层；

③ 算术表达式太复杂，括号嵌套超过36层；

④ LOMEM：和HIMEM：两语句所定义的变量值不在－65535 到65535 之间。

(10) ? OVERFLOW ERROR 打键输入或程序计算产生的数字太大，发生溢出。

(11) ? REDIM'D ARRAY ERROR重新定义数组变量错。若是程序中对于已定义了的数组变量又去重复定义则出错。

(12) ? RETURN WITHOUT GOSUB ERROR 指在RETURN之前没有与之对应的GOSUB语句。

(13) ? STRING TOO LONG ERROR字符串太长错。字符串表达式的长度超过255个字符时出错。

(14) ? SYNTAX ERROR语法错。所用语法不对，或其它出错信息不能表示的出错。

(15) ? TYPE MISMATCH ERROR类型不匹配错。语句中提供的值（数或字符串）与变量的类型不匹配。

(16) ? UNDEF'D FUCTION ERROR未定义函数错。程序中调用了一个还设有定义的函数。

(17) ? UNDEF'D STATEMENT ERROR 未定义语句错。程序中用GOTO或GOSUB语句转向一个不存在的行号时出错。

以上的出错信息可在222 内存单元得到错误编码，如下表：

PEEK (222)	出错情况
0	NEXT与FOR不配对
16	语法错
22	RETURN与GOSUB不配对
42	数据不够用
53	非法数值
69	溢出

77	内存不够用
90	未定义语句
107	数组变量定义不正确
120	重复定义数组变量
133	分母为零
163	类型不匹配
176	字符串太长
191	公式太复杂
224	用到的函数未定义
254	响应INPUT语句时，输入不适当
255	企图作CTRL－C中断

§ 2.11 DOS命令

2.11.1 文件名称与类型

计算机应用中所指的文件是存放在外部存贮设备（如磁盘）中的一批数据，在一般意义上说，某一名称文件内的信息的目的、形式和内容上彼此相似。这类数据既可以是字符，也可以是数字，或兼而有之。所以文件可以代表一段文字，也可能表现为一段程序或一批数字。每个文件都必须起一个名字，称为文件名，根据磁盘操作系统（DOS）命令可对文件存取或管理。文件取名的规则是：

（1）文件名的长度必须在1到30个字之间，超出的字将无效而被舍掉。

（2）文件名必须由字母开始。

（3）任何键盘输入的字符都可以作为文件名的一部分，但是逗号不能用。

(4) 也可以用不会显示在屏幕上的字符（例如有些与CTRL键合用时）做为文件名的一部分，在磁盘目录上也不显示。因此一般不将控制字符插入到文件名中，当然这也不失为使文件名保密的方法之一。

APPLE Ⅱ 的DOS 操作系统提供某些命令来定义执行时的操作，CATALOG命令可将磁盘上文件名输出（显示），所显示目录的最左一行表示文件类型，其存取和管理方法均有所不同。显示代码所表示的文件类型如下：

代 码	意 义
A	APPLESOFT程序
B	二进制映象文件
I	整型BASIC程序
T	文本文件
R	可被执行的二进制文件

2.11.2 指定磁盘驱动器

前面已经介绍过APPLE Ⅱ PLUS 微型计算机有自动引入磁盘操作系统（DOS）的功能，但是开机之前必须把系统主盘插在与驱动器插件的DRIVE 1 接口相联的驱动器（称 1 号驱动器）中，然后开机，这时这个驱动器的红灯点亮，显示屏上渐渐显示磁盘操作系统引入的信息。过一会儿，驱动器红灯灭，显示屏出现APPLESOFT 提示符，即可操作了。在有两个驱动器的情况下，1号驱动器命名为D1；另一驱动器命名为D2。驱动器工作状态有如下特点：键入DOS命令（后面介绍）时若没有指定驱动器号，则表示这个命令将访问 1 号驱动器。若要访问 2 号驱动器，则在DOS命令的后面要有指定 2 号驱动器的参数。例如若是开机后立即发出CATALOG命令，则表示列出D 1 驱动器中磁盘目录，

若键入CATALOG，D2,则表示列出D 2 驱动器中目录。但在D 2 驱动器被指定后，若不再重新指定D 1 驱动器，则所有磁盘命令将访问D 2 驱动器。

2.11.3 初始化命令

刚买来的新磁盘是不能立即用作存取盘的，厂家生产的磁盘是适应不同的计算机用的。到具体的计算机上使用之前，先得在该计算机上做初始化（INITIALINE)工作,也称格式化，即：

（1）对磁盘进行格式化。若这个磁盘不是才买来的新盘，而是存有资料的旧盘，则还会抹掉上面的所有信息，然后划分磁道和扇区的地址。

（2）将DOS存放在磁盘的$00—$02三个磁道上，因此，凡是已经格式化的磁盘都可以作为DOS的引导盘了。

（3）为磁盘建立文件目录区（第$11磁道),填写该磁盘的卷号，卷号可以在初始化时指定，若不指定，则为254号。一个磁盘只有一个卷号，并不能修改。

（4）在发出初始化命令之前,主机内存中已有的BASIC程序将作为这个磁盘以后用于开机自动调入执行的第一个程序或称问候程序，或称欢迎程序。因为初始化后，这个欢迎程序以文件形式存在磁盘上了，并附以文件名称。若是在初始化之前没有特殊必要，可用原有DOS引入后主机中的程序作为欢迎程序、不再设计特别的程序，即可键入初始化命令；若有特殊需要，也可以先设计一个欢迎程序，然后初始化。例如，假如要在一块新盘上存一批数学方法程序，为使将来这块盘做引导盘时,可立即在屏幕上显示它的专门用途，则在这块盘初始化之前，先在主机中键入如下程序：

```
10   HOME
20   VTAB 10
30   PRINT  SPC( 8);"MATHMATICAL M
      ETHOD"
50   VTAB 20
60   PRINT  SPC( 23);"MET.INS.of  "

70   PRINT
80   PRINT  SPC( 25);"JILIN PROVIN
      CE"
```

然后再键入初始化命令。将来当这块盘用作引导盘时，引入DOS后将立即自动运行这个欢迎程序，并在屏幕上有如下显示：

```
MATHMATICAL METHOD



MET.INS.of

  JILIN PROVINCE
```

磁盘格式化的方法：

①若只用一个磁盘驱动器

步骤：

a）将系统主盘插入D 1 驱动器，开机；

b）若有特殊需要可在主机打入欢迎程序，若无这个要求，则可不作处理。拿出主盘，放进新盘；

c）键入命令INIT HELLO回车。其中HELLO是欢迎程序文件名。这时驱动器红灯亮，发出转动声，约两分钟后驱动器红灯灭，格式化完成。

② 若用两个磁盘驱动器

步骤：

a）将系统主盘插入D1驱动器，要做格式化的新盘插入D 2 驱动器，开机；

b）打入欢迎程序，或不作处理；

c）键入命令INIT HELLO,D 2，回车，驱动器红灯亮，发出转动声，约两分钟，驱动器红灯灭，格式化完成。

格式化后，磁盘上已存有名字为HELLO的欢迎程序，当这个磁盘将来用作引导盘时，将立即自动运行这个欢迎程序。已经格式化的磁盘应贴上标签注明在多大内存容量系统中进行初始化，以及初始化日期。例如“48K盘1984年 1 月 1 日初始化。”

2.11.4　磁盘格式

每个磁盘被APPLE Ⅱ 的 DOS分成35个同心圆（磁道，track)，这些磁道编号为 0 到34($ 0 到$ 34)，每一个磁道分为16个扇区（sector)，扇区编号为 0 到15（$ 0 到 $ F)，每个扇区可以贮存256个字节资料。所以在一个磁盘上共有 560 个扇区，共可贮存143,360个（143K)字节的资料。其中DOS占用 3 个磁道48个扇区位置，磁盘索引文件占用 1 个磁道16个扇区的位置，其余31个磁道共 496 个扇区用于贮存资料和程序文件。

第11个磁道做为磁盘的索引文件位置，这里对每个文件

列出了名称、文件形式、占用扇区数目以及是否锁住等信息。索引文件磁道的每一个扇区最多可以存放七个文件的资料，整个索引文件位置最多可容纳105个文件的索引资料（第0扇区不存贮索引文件）。

2.11.5　整片磁盘的复制

把系统主盘（DOS 3.3版本）插入D1驱动器，键入CATALOG(列目录命令)，就能把主盘上的文件全部列出如下：

```
DISK VOLUME 254

*A 006 HELLO
*I 018 ANIMALS
*T 003 APPLE PROMS
*I 006 APPLESOFT
*I 026 APPLEVISION
*I 017 BIORHYTHM
*B 010 BOOT13
*A 006 BRIAN'S THEME
*B 003 CHAIN
*I 009 COLOR DEMO
*A 009 COLOR DEMOSOFT
*I 009 COPY
*B 003 COPY.OBJO
*A 009 COPYA
*A 010 EXEC DEMO
*B 020 FID
*B 050 FPBASIC
*B 050 INTBASIC
*A 028 LITTLE BRICK OUT
*A 003 MAKE TEXT
*B 009 MASTER CREATE
*B 027 MUFFIN
*A 051 PHONE LIST
*A 010 RANDOM
```

```
*A 013 RENUMBER
*A 039 RENUMBER INSTRUCTIONS
*A 003 RETRIEVE TEXT
```

其中DISK VOLUME 254表示磁盘为254卷； *符号代表文件已锁住(见后面介绍)； I等字符代表文件形号；三位数字代表这个文件所占用的扇区，最小的文件(空文件)占有一个扇区。若一个文件占用的扇区超过255个，则这个数目将从零开始重新计算，但这并不影响这个文件的真正占用位置。

以下逐个介绍各个文件的简要含义，详细的应用将在后面章节中分别介绍。

HELLO文件是欢迎程序，为APPLESOFT 型文件，占6个扇区。开机即自动运行显示，并把INTEGER解释程序装入内存。

ANIMALS 是游戏程序，只有去掉写保护后才能顺利执行，因为执行过程中有写入磁盘的操作。

APPLESOFT 用于并非自启动固化ROM的开机程序，对APPLE II PLUS无效。

APPLEVISION 游戏程序，运行唱歌跳舞程序。

BIORHYTHM 游戏程序，生命曲线演示。

BOOT13 13扇区磁盘的引导程序。

CHAIN 程序连接运行程序。

COLOR DEMO 用整型 BASIC写的彩色示范程序。

COLOR DEMOSOFT 用APPLESOFT BASIC写的彩色示范程序。

COPY 复制整片磁盘程序，用整型BASIC编写。

COPYA 用APPLESOFT BASIC编写的复制整片磁盘程序。

EXEC DEMO 用EXEC命令的示范程序。

FID 多种功能的程序。

FPBASIC为APPLESOFT的解释程序。

INTBASIC 整型BASIC的解释程序。

LITTLE BRICK OUT 用游戏操纵杆玩的游戏程序。

MAKE TEXT 建立文本文件的示范程序。

MASTER CREATE 建立启动系统的应用程序。

MUFFIN 将13扇区磁盘转换成16扇区程序。

PHONE LIST 存贮电话号码的程序。

RANDOM 随机文件的示范程序。

RENUMBER 重编程序号和处理程序连接的程序。

RENUMBER INSTRUCTIONS 重编号程序使用的说明。

RETRIEVE TEXT 检索文本文件程序。

由于系统主盘对微型机的使用很是重要，除应妥善保管外，还应该复制供平时使用，原盘更应很好保管。在这里先介绍整片磁盘复制的方法（复制前新磁盘不必先做格式化工作）。

只用一个驱动器复制磁盘的方法

步骤：

① 把系统主盘（DOS3.3版本）插入驱动器；

② 键入RUN COPYA，回车，这时从磁盘导入程序并运行，在屏幕上显示：

APPLE DISKETTE DUPLICATION PR-

OGRAM

ORIGINAL SLOT: | DEFAULT=6 |

表示已进入复制程序，要求指定“源盘”的驱动器插件槽号，同时在右边以缺席选择显示提醒槽号为6。在这种情况下若操作人员不作选择，即表示系统认为驱动器已插在第6个槽号上了。若我们已肯定插在第6个槽号，则只要简单地按回车键就行了，这时又显示：

ORIGINAL SLOT: 6

DRIVE: | DEFAULT=1 |

表示“源盘”的驱动器插件已确定插在6号槽，要求指定“源盘”的驱动器编号，在右边以缺席选择显示提醒驱动器号为1。若我们肯定“源盘”已在1号驱动器，则也只要按回车键就行了，这时又显示：

DRIVE: 1

DUPLICATE SLOT: | DEFAULT=6 |

表示“源盘”的驱动器已指定为1号，要求指定“复制盘”的驱动器插件号，右边以缺席选择显示提醒槽号是6，这时再按回车键，则显示：

DUPLICATE SLOT: 6

DRIVE: | DEFAULT=2 |

表示已确认“复制盘”的插槽是6号，这时又要求输入“复制盘”的所在驱动器号，右边提示为2号驱动器，但是由于我们只用一个驱动器，“复制盘”在复制时也必须放在1号驱

动器，所以我们只好按键1，这时立即显示：

--PRESS ‘RETURN’ KEY TO BEGIN COPY——要求按回车键进入复制。按回车键，则显示：

INSERT ORIGINAL DISK AND PRESS RETURN要求插入“源盘”并按回车键。照此办理(若是复制系统主盘,则不必把主盘拿出来了；若复制别的“源盘”，则必须拿出系统主盘，插入“源盘”)，则在显示屏的“源盘”槽口右边位置显示READING，驱动器红灯亮，不久红灯灭，显示：

INSERT DUPLICATE DISK AND PRESS RETURN 要求取出源盘，把“复制盘”插入驱动器，再按回车键。照此办理，这时在显示屏复制盘槽口指示位置又以缺席选择显示FORMATING，表示在对“复制盘”格式化，灯亮。不久灯灭，显示：

INSERT ORIGINAL DISK AND PRESS RETURN又要求插入源盘，按回车键。照此办理又显示：READING，灯亮，灯灭又显示：

INSERT DUPLICATE DISK AND PRESS RETURN 又要求换插“复制盘”，按回车键。这时屏幕上显示：WRITING

这类过程得反复多次，即在“源盘”上读一段，换盘，在“复制盘”上写一段。最后显示：

DO YOU WISH TO MAKE ANOTHER COPY?询问你是否还复制一个盘，若键入Y(意即YES)，表示要求再复制一个盘，则重复上述复制的全过程；若键入N(意即NO)，则表示脱离复制操作，又回到提示符。

用两个驱动器复制磁盘的方法

从上面的操作看到，用一个驱动器复制盘较麻烦，用两个驱动器复制则方便得多。

步骤：

① 系统主盘插入1号驱动器，“复制盘”插入2号驱动器

② 键入RUN COPYA回车。这时从磁盘导入程序并运行，在屏幕上显示：

APPLE DISKETTE DUPLICATION PROGRAM

ORIGINAL SLOT: | DEFAULT = 6 |

若确认为6号槽，则按回车，又显示：

ORIGINAL SLOT: 6

DRIVE: | DEFAULT = 1 |

若确定为1号驱动器，则按回车，又显示：

DRIVE: 1

DUPLICATE SLOT: | DEFAULT = 6 |

若确认也是6号槽口，则按回车键，显示：

DUPLICATE SLOT: 6

DRIVE: | DEFAULT = 2 |

若“复制盘”已插入2号驱动器，则按回车键并显示：

——PRESS ‘RETURN’ KEY TO BEGIN COPY——要求按回车键。

③ 若是复制系统主盘，则就可执行复制了；若是复制别

的“源盘”,则必须把“源盘”换入1号驱动器，然后按回车键，这时屏幕上交替显示READI NG, FORMATING, READING, WRITING等，最后显示：

DO YOU WISH TO MAKE ANOTHER COPY? 询问你是否再拷贝一个盘，若键入Y，则继续像上述执行复制；若键入N，则退出复制。

在复制软盘过程中如出现“******UNABLE TO READ ******”则表示“源盘”已坏不能复制。

2.11.6 DOS 命令

前面§2.2已经介绍了磁盘操作系统（DOS）的引入及版本等内容，这一节介绍它的用途和命令。所谓DOS实际上是主机通过磁盘驱动器自动追索磁盘文件磁道、存储和检索信息，并处理大量其它管理事务的程序。BASIC将任何对磁盘作业的要求传送到DOS，而DOS将结果传回给BASIC。一个新的磁盘启用前，必须先做初始化工作（见§2.11.3）,这样DOS就写在0，1和2号磁道上了。每次开机，都要引入DOS，这时，才可根据需要执行各种DOS命令。

DOS 命令的一般格式是：

DOS 命令　文件名，[Ss]，[Vv]，[Dd]

其中文件名限定长度只能是1到30个字符，第一个字符必须是字母，文件名中不能用逗号，CTRL-M或RETURN；参数S是指驱动器接口的插槽号，例如S 7指插在第7号插槽的驱动器，一般驱动器接于插在6号插槽的插件上，DOS就在这个插槽的驱动器引入，故用到的DOS命令中这个参数一般就省略掉了；参数V指的是磁盘的卷号，磁盘的卷号为1—254的整数，每个磁盘的卷号在INIT命令中就

指定了，若是INIT命令中没有指定卷号，则分配给磁盘的卷号为254，为使用方便起见，常常不指定卷号，故这个参数也常常省略；参数D指的是驱动器号，若只用两个驱动器，则只有D1和D2。以上DOS命令格式中可选项可以任意次序排列。

以下介绍主要的DOS命令：

CATALOG

这是列出磁盘文件目录的命令，一般格式是：

CATALOG [Ss], [Vv], [Dd]

其中参数Ss，Vv和Dd根据需要选用。根据前面说过的理由，一般选用参数Dd就可以了。例如：

CATALOG, D2

即为指定显示D2驱动器中的文件目录。

LOAD

是装入命令，一般格式是：

LOAD文件名，[Ss]，[Vv]，[Dd]

从磁盘读入程序文件到主机内存，命令后面必须跟着文件名，一般后面还可指定驱动器号。例如：

LOAD HELLO, D1

表示从1号驱动器把HELLO文件装入主机。若磁盘中没有这个文件，则主机会发出“嘟”的一声，以示警告，并显示FILE NOT FOVND（没有找到文件）；若磁盘上有这个文件名称，但不是一个程序文件，则也发出“嘟”声，显示FILE TYPE MISMATCH（文件类型不对）；假如一切顺利，则这个命令会把主机内的原有程序清除掉，而装入这个新文件。

RUN

是运行命令，一般格式是：

RUN 文件名，[Ss]，[Vv]，[Dd]

例如：

RUN ABC，D2

即先从磁盘装入程序ABC，并立即运行，指定驱动器号D2。

SAVE

是存入命令，一般格式是：

SAVE 文件名，[Ss]，[Vv]，[Dd]这个命令把主机内程序保存到磁盘中，作为该程序的副本。例如把程序ABC存入驱动器D2中磁盘上，应键入：

SAVE ABC，D2

SAVE命令结束后，可以用CATALOG命令来检查程序是否已经复制完成。

DELETE

是删除命令，一般格式是：

DELETE 文件名，[Ss]，[Vv]，[Dd] 这个命令把磁盘上指定的文件删除。例如：

DELETE XY，D1

表示把1号驱动器磁盘的XY文件删除。

LOCK

是锁定命令，一般格式是：

LOCK 文件名，[Ss]，[Vv]，[Dd]这个命令锁定磁盘上的文件。例如：

LOCK ABC，D2

文件锁定后，对磁盘上的文件起了保护作用，这时只能对这个文件执行LOAD或READ（见后面介绍的数据文件操作），而无法执行SAVE，DELETE或WRITE(数据文

件操作)，因而这个文件不致于被无意冲掉，起到了保护作用。锁住的文件在磁盘目录栏以*标记。

UNLOCK

是开锁命令，一般格式：

UNLOCK文件名，[Ss]，[Vv]，[Dd]

例如：

UNLOCK ABC，D2

用了开锁命令之后，又可对文件进行SAVE，DELETE，WRITE 操作了。磁盘目录中*标记消失。

RENAME

是改换磁盘上的文件名的命令，常用格式是：

RENAME 文件的旧名，文件的新名，Dd 用这个命令时必须注意，因为DOS 操作系统不检查这个新名字在磁盘上是否早已有了。因此，新命名必须避免重复。不然将得到同名的两个文件，就难办了。

VERIFY

是检验磁盘上文件是否完整的命令。常用格式：

VERIFY 文件名，Dd （d=1或2）

这个文件可检查文件存到磁盘上去过程中是否正确，所占磁盘区是否有硬伤。DOS 3.3版本在存贮文件完毕后都自动执行VERIFY操作。若VERIFY操作后显示I/OERROR，说明存储文件出错，应该重新存盘。

2.11.7 RENUMBER 程序

系统主盘上的RENUMBER 程序文件有两个功能：改变程序行号；把两个程序连接起来，一般用于把原先存贮于磁盘上的几个程序根据需要连接起来。

RENUMBER 程序用法：

引入RENUMBER文件

把系统盘插入驱动器，键入RUN RENUMBER 命令，回车，则屏幕显示：

```
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&                                      &
&          APPLESOFT RENUMBER          &
&                                      &
&  COPYRIGHT 1978  APPLE COMPUTER INC  &
&                                      &
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

RENUMBER   (DEFAULT VALUES)

& [FIRST 10] [,INC 10] [,S 0] [,E 63999]


MERGE

&H    PUT PROGRAM ON HOLD
&M    MERGE TO PROGRAM ON HOLD



PRESS 'RETURN ' TO CONTINUE...
```

表示将进入RENUMBER程序，并说明这个程序的用法，最后要求按回车键。按了回车键后，屏幕显示：

```
    RENUMBER IS INSTALLED AND READY

  IF YOU USE 'FP', 'HIMEM', OR 'MAXFILES'

   YOU WILL HAVE TO RE-RUN RENUMBER
```

说明RENUMBER文件已进入内存可以使用，假如今后操作过程中用到了FP，HIMEM或MAXFILES则将把这个程序从内存中冲掉。要指出的是，由于RENUMBER程序文件进入内存，所以这时主机可用内存已减少。

改变程序行号的方法

改变行号的格式为：

&F起始行号，I行号间隔

把RENUMBER程序引入主机后，即可把需改变行号的程序引入主机，例如有如下程序：

```
10   FOR I = 1 TO 15
20   PRINT I
30   NEXT I
```

若要改变行号为100行开始，2行间隔，则应该键入&F100，I2回车。操作和结果如下：

```
]&F100,I2

]LIST

100  FOR I = 1 TO 15
102  PRINT I
104  NEXT I
```

把程序连接起来的方法

主机内存中有了RENUMBER程序后，就能把几个程序（一般逐个从磁盘调入）连接起来。方法是键入命令&H把主机中现有程序保护起来，然后从磁盘引入第二个程序

(LOAD文件名)，再键入&M就可把两个程序连接起来。例如在磁盘上有两个文件A和B，要把这两个程序连接起来，其中还把一个程序的行号改变了，以组成完整的程序，整个操作如下：

```
üLOAD A,D2
üLIST

10   DIM A(15)
20   FOR I = 1 TO 15
30   READ A(I)
40   NEXT I

ü&H
PROGRAM ON HOLD, USE "&M" TO RECOVER

üLOAD B
üLIST

10   DATA 1,3,5,7,9
20   DATA 2,4,6,8,10
30   DATA 3,6,9,12,13

ü&F100,I10

üLIST

100  DATA 1,3,5,7,9
110  DATA 2,4,6,8,10
120  DATA 3,6,9,12,13

ü&M

üLIST

10   DIM A(15)
20   FOR I = 1 TO 15
```

```
30 *READ A(I)
40  NEXT I
100  DATA 1,3,5,7,9
110  DATA 2,4;6,8,10
120  DATA 3,6,9,12,13
```

2.11.8 顺序处理文件

APPLE的DOS可以提供两种类型的磁盘文件，即顺排文件及随机存取文件。顺排文件在存贮器中按线性顺序排列，并以其物理顺序存取，也可称为顺序处理文件，而随机存取文件则在存取时可不考虑记录存放的排列次序。现分别介绍如下。

顺排文件为文本(TEXT)文件之一种，用CATALOG命令显示文件目录时，将伴随有"T"的标记。建立这类文件必须通过程序编制和运行才行。成批的数据或字符可以按顺排文件方式存储到磁盘上，方便今后调出使用。

顺序处理文件在磁盘上的存贮格式

顺序处理文件中字符是用ASCII码来代表的，每个字符在文件中按次序安排，没有空位隔开，每组字符用回车符隔开（下图以↲代表回车符），每个字符是1个字节，每组字符为1个字段，结构图如下：

顺序处理文本文件格式个例图

字符:	7	↲	A	T	↲	O	N	E	↲	B	L	O	W	↲					
ASCII码	55	13	65	84	13	79	78	69	13	66	76	79	87	13	00	00	00	00	00
文件字节	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
字段	0		1			2				3									

在顺序处理文本文件的存取中就必须考虑到字节和字段的安排。

在程序中使用DOS命令

DOS 命令除了由键盘输入外，也可以在程序中使用，用于处理文件，进行主机与磁盘交换信息。方法是在程序中放置控制字，有两种方法：

PRINT "CTRL-D"; "DOS 命令"

或PRINT CHR$(4); "DOS 命令"

其中CTRL-D指的是同时按CTRL键和D键，得到的字符在屏幕上是不显示的，就象空格一样。CTRL-D的实际结果是形成ASCII码4，在程序中为了使用方便，常常把这个控制字定义为一个字符串变量。例如若要求列磁盘文件目录，则程序设计如下：

```
10  D$=CHR$(4):REM  CHR$(4)=CTRL
20  PRINT  D$;"CATALOG,D2"
30  END
```

程序第10行设置了控制字，并将CTRL-D控制字定义为D$。第20行发出列目录命令前加上D$，就代表了控制字。运行这个程序就把2号驱动器中磁盘文件目录显示出来。

程序中使用其它DOS 命令的方法也类似。

用程序控制打印机的方法也与此类似。因一般在操作中已将DOS 引入主机，所以PR# 就象是一个DOS 命令，在程序中也必须与控制字联用，所以程序控制打印机的格式是：

```
10  D$=CHR$(4):REM  CHR$(4)=CTRL-D
```

```
20 PRINT D$:"PR#1"
30 PRINT 有关资料
40 PRINT D$:"PR#0"
```

以上程序中，第10行建立控制字CTRL-D，第20行打开打印机，第30行打印语句把资料打印列出，第40行关闭打印机。

若DOS不在主机内存中，则程序控制打印机的格式是：

```
10 PR#1
20 PRINT 有关资料
30 PR#0
40 END
```

磁盘监控命令

用程序在磁盘上建立文本文件或取文件时，在显示屏上没有什么显示，为了能从屏幕上监督这类操作，可以使用磁盘监控命令MON。它的格式是：

MON C，I，O

其中C表示对磁盘操作命令(如OPEN,READ等)的监控；I表示对由磁盘输入（INPUT)监控；O代表对输出(OUT)到磁盘的监控。这三个参数可以进行以下排列组合方式，若MON命令没有参数，命令将不执行。

① MON C 只对磁盘操作命令的监控；

② MON I 只对由磁盘输入的情形监控；

③ MON O 只对输出到磁盘的情形监控；

④ MON C，I 监控磁盘命令和由磁盘的输入；

⑤ MON C,O 监控磁盘命令和向磁盘输出；

⑥ MON I，O 监控由磁盘的输入和向磁盘输出；

⑦ MON C，I，O　全部监控

为了退出监控，可使用解除监控的命令NOMON，类似的它也有以上7种对应的用法。

监控命令可作为键盘直接输入命令使用，也可以用到程序的语句中去，但是在它前面也必须有控制字，例如：

10　PRINT CHR$(4)；“MON C，I，O”

主机进入监控命令后就一直保持有效，直到执行NOMON，INT，FP命令或CTRL-RESET（强制复位）才解除。

顺序处理文件用法

① 存取字符的顺序处理文件

a）建立字符串的文本文件程序

下面这个程序是用来建立一个文本文件的，内存两个字符串ABC以及123，各自占一个字段。运行这个程序（用监控命令显示）就能看到它建立文本文件的全过程：先是打开一个文件，名字是WORDS 1，然后准备写进这个文件中的内容（即字符串ABC及123），各占一个字段，最后关闭这个文件。文件建完后列出磁盘目录，看是否已增加了名称为WORDS 1的文件。程序及建文件过程如下：

```
10  REM MAKE WORDS1
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN WORDS1"
40  PRINT D$;"WRITE WORDS1"
50  PRINT "ABC"
60  PRINT "123"
70  PRINT D$;"CLOSE WORDS1"
100  END

]MONC,O
```

```
]RUN
OPEN WORDS1
WRITE WORDS1
ABC
123
CLOSE WORDS1
```

若是预先未设置监控命令，建文件过程将不在屏幕上显示。存取文件程序要注意在程序中须有控制字，DOS 命令(OPEN，WRITE 等)必须与控制字联用。

b) 从文件中取字符串的程序

从文件中取字符串的程序先要打开相应的数据文件，给予读命令，再给予读取语句，最后关闭这个文件。程序及监控运行结果如下：

```
10  REM RETRIEVE WORDS1
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN WORDS1"
40  PRINT D$;"READ WORDS1"
50  FOR I = 1 TO 2
60  INPUT A$(I)
70  NEXT I
80  PRINT D$;"CLOSE WORDS1"
100  END

]MON C,I

]RUN
OPEN WORDS1
READ WORDS1
?ABC
?123
CLOSE WORDS1
```

以上程序设计要点如下：必须知道存有字符串的数据文件内字符串安排的格式；DOS命令必须与控制字联用；文件先打开，后读取，最后关闭文件。CLOSE命令的后面跟随文件名表示这个命令关闭该文件，若CLOSE后面不跟具体的文件名，即表示关闭所有已打开的文件。

c）建立另一种形式存入字符串文件的程序

即以程序方式将字符串ABC与123存入到一个字段中去。程序及监控运行情况如下：

```
10  REM MAKE WORDS2
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN WORDS2"
40  PRINT D$;"WRITE WORDS2"
50  PRINT "ABC,123"
60  PRINT D$;"CLOSE WORDS2"
100  END

]MON C,I,O

]RUN
OPEN WORDS2
WRITE WORDS2
ABC,123
CLOSE WORDS2
```

这个数据文件的名字是WORDS2。

d）按以上格式存入字符串的文件取出程序

针对上述字符串在文件中的存贮格式，取这种字符串的程序与运行情况如下：

```
10 .REM RETRIEVE WORDS2
20 D$ =  CHR$ (4): REM CTRL-D
```

```
30   PRINT D$;"OPEN WORDS2"
40   PRINT D$;"READ WORDS2"
50   INPUT A$,B$
60   PRINT D$;"CLOSE WORDS2"
100   END

]MON C,I,O

]RUN
OPEN WORDS2
READ WORDS2
?ABC,123
CLOSE WORDS2
```

这时A＄代表字符串ABC，B＄是字符串123；注意不能用以下两个程序行：

```
50 INPUT  A$
60 INPUT  B$
```

代替上述第60行程序，防止运行出错。

以上介绍了两对存取顺序处理文件的例子。在将数据存入文件时需注意磁盘上是否已有了同样名称的文件，资料存入名称相同的文件会造成混杂。解决的办法是在建立新文件前，应先把老文件删掉。但假如原来磁盘上没有同样名字的文件，执行删除命令，也会出错的。为了避免出错，可以在程序中先打开这个文件，(打开不存在的文件是不出错的)，然后删去，再打开文件，然后执行写命令。程序格式如下：

```
10 D$ =CHR$ (4):REM  CTRL-D
20  PRINT D$;"OPEN 文件名"
30 PRINT D$;"DELETE 文件名"
40 PRINT D$;"OPEN 文件名"
```

```
50 PRINT D$;"WRITE 文件名"
60 PRINT  有关资料
70 PRINT D$;"CLOSE"
80 END
```

因此与前面列举的两个建文件的程序类似的程序，应添加打开—删除—打开—写这样的程序段。

② 存取数据的顺序处理文件

a）把一批数据存入磁盘

以下程序把10个数据顺序存入磁盘，并监控了建文件过程。

```
5  DIM A(10)
10 D$ = "": REM CTRL-D
20  PRINT D$;"OPEN EX102"
30  PRINT D$;"DELETE EX102"
40  PRINT D$;"OPEN EX102"
50  PRINT D$;"WRITE EX102"
60  FOR I = 1 TO 10
70  READ A(I)
80  PRINT A(I)
90  NEXT
100  PRINT D$;"CLOSE"
110  END
120  DATA 50,60,70,80,90,100,110,
     120,130,140
```

运行结果：

```
ÜMON C,I,O

ÜRUN
```

```
OPEN EX102
DELETE EX102
OPEN EX102
WRITE EX102
50
60
70
80
90
100
110
120
130
140
CLOSE
```

以上程序运行结果在磁盘上建立了数据文件EX102，由于已有删除旧文件命令，避免了新旧资料的混杂。

b）从顺序处理文件中取数据

针对已建的数据文件，设计了如下取数据程序，运行后即把数据取出。程序及监控运行过程如下：

```
5  DIM A(10)
10 D$ = "": REM CTRL-D
20  PRINT D$;"OPEN EX102"
30  PRINT D$;"READ EX102"
40  FOR I = 1 TO 10
50  INPUT A(I)
60  NEXT
70  PRINT D$;"CLOSE"
80  FOR I = 1 TO 10
90  PRINT A(I),
100  NEXT
110  END
```

```
]MON C,I,O

]RUN
OPEN EX102
READ EX102
?50
?60
?70
?80
?90
?100
?110
?120
?130
?140
CLOSE                60                70
50                   90                100
80                   120               130
110
140
```

程序中所用INPUT语句用于读一个字段，各数据赋予语句中各变量。

c）文件名的间接用法

为了增加程序使用上的灵活性，所存取的文件名可使用间接名称。例如下面这个取文件数据的程序及监控运行结果：

```
5   DIM A(10)
10  D$ = "": REM CTRL-D
20  INPUT "FILE NAME=";F$
25  PRINT D$;"OPEN";F$
30  PRINT D$;"READ";F$
35  FOR I = 1 TO 10
40  INPUT A(I)
```

```
50   NEXT
60   PRINT D$;"CLOSE"

]MON C,I,O

]RUN
FILE NAME=EX102
OPENEX102
READEX102
?50
?60
?70
?80
?90
?100
?110
?120
?130
?140
CLOSE
```

③ 往顺序处理文件中增加资料

建立顺序处理文件时，以OPEN命令打开文件，此时，文件读/写定位指针将设在文件的第一个字符上(即零字节)。APPEND命令则能把文件打开，并将文件读/写定位指针设到文件最后一个字符后面一个字节上。所以用APPEND命令打开一个旧文件后，就能往里面增加资料了。例如通过以下程序建立了一个数据文件，名称为EX201(间接名称为F$)：

```
10 D$ =  CHR$ (4): REM CTRL-D
20  INPUT "FILE NAME=";F$
30  PRINT D$;"OPEN";F$
```

```
40   PRINT D$;"DELETE";F$
50   PRINT D$;"OPEN";F$
60   PRINT D$;"WRITE";F$
70   FOR I = 1 TO 10
80   PRINT I
90   NEXT
100   PRINT D$;"CLOSE"
```

可通过以下程序在已建立了的原数据文件之后增加数据：

```
10 D$ =  CHR$ (4): REM CTRL-D
20  INPUT "FILE NAME=";B$
30  PRINT D$;"APPEND";B$
40  PRINT D$;"WRITE";B$
50  FOR I = 1 TO 10
60  PRINT I * I
70  NEXT
80  PRINT D$;"CLOSE"

]MON C,I,O

]RUN
FILE NAME=EX201
APPENDEX201
WRITEEX201
1
4
9
16
25
36
49
64
81
100
CLOSE
```

通过以下程序，可从数据文件E X 201 中取出两个程序所存入的原存数据和增加数据。

```
5  DIM A(20)
10  INPUT "FILE NAME=";C$
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN";C$
40  PRINT D$;"READ";C$
50  FOR I = 1 TO 20
60  INPUT A(I)
70  NEXT
80  PRINT D$;"CLOSE"
90  FOR I = 1 TO 20
100  PRINT A(I),
110  NEXT

]MON C,I,O

]RUN
FILE NAME=EX201
OPENEX201
READEX201
?1
?2
?3
?4
?5
?6
?7
?8
?9
?10
?1
?4
?9
?16
?25
?36
?49
?64
```

```
?81
?100
CLOSE
1                  2                  3
4                  5                  6
7                  8                  9
10                 1                  4
9                  16                 25
36                 49                 64
81                 100
```

④ 从任意字段开始读写数据

命令：POSITION文件名[，Rp]

这个命令把文件读/写定位指针设在第p个字段上。若p=0，则读、写命令从现在字段开始；若p=1，则从下一个字段开始；若p=2，则从距现在字段的第二个字段开始，依此类推。但若POSITION命令所设定字段中已无数据，则读时会出错。下面的例子是从已建立的EX201中取一些数据：

```
10  REM POSITION FILE
20  INPUT "FILE NAME=";A$
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN";A$
50  PRINT D$;"POSITION";A$;",R2"
60  PRINT D$;"READ";A$
70  INPUT X
80  PRINT D$;"POSITION";A$;",R1"
90  PRINT D$;"READ";A$
100  INPUT X
110  PRINT D$;"OPEN";A$
120  PRINT D$;"POSITION";A$;",R16
     "
```

```
130  PRINT D$;"READ";A$
140  INPUT Y
150  INPUT Z
160  PRINT D$;"CLOSE"

]MON C,I,O

]RUN
FILE NAME=EX201
OPENEX201
POSITIONEX201,R2
READEX201
?3
POSITIONEX201,R1
READEX201
?5
OPENEX201
POSITIONEX201,R16
READEX201
?49
?64
CLOSE
```

由于磁盘操作中的INPUT命令一方面把磁盘上的数据字段读入主机，而又根据POSITION命令把文件定位指针推进到所指定的字段的开始，所以有以上的运行结果。

⑤ 从任意字节开始读写数据

对顺序处理文件可以从任意字节开始读写数据，要注意之点是：

a) 每个字符相当于一个字节，在存取时应该确切知道字节位置；

b) 第一个字节编号为零，字符、空格、逗号和RETURN都是字节（RETURN是一个字节）；

c）存数据格式：

WRITE文件名，Bb

读数据格式：

READ文件名，Bb

其中B为参数，若不指定B，B就是零，即从文件中第一个字节开始。以下是一个从上面已建立的文件中读数据的程序以及运行的例子：

```
10  REM BYTE READ
20  INPUT "BYTE NO.=";P
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN EX201,D2"
50  PRINT D$;"READ EX201,B";P
60  INPUT X
70  INPUT Y
80  INPUT Z
90  PRINT D$;"CLOSE"

]MON C,I,O

]RUN
BYTE NO.=27
OPEN EX201,D2
READ EX201,B27
?16
?25
?36
CLOSE
```

第30行置控制字，第40行指定2号驱动器，打开文件EX201（以上例子中建立了这个文件，内存20个数），第50行从指定字节p开始读数据，60—80行读进3个字段。p=27时，X=16，Y=25，Z=36

再运行一次：

```
]RUN
BYTE NO.=26
OPEN EX201,D2
READ EX201,B26
?
?REENTER
?16
?25
?36
CLOSE
```

p=26时，正好是RETURN字节，因此读数不正常，但还是继续执行结束了，仍取到3个字段的数据。

再运行一次：

```
]RUN
BYTE NO.=28
OPEN EX201,D2
READ EX201,B28
?6
?25
?36
CLOSE
```

p=28时，是一个字段的部分字节处开始，第一个字段只取得部分字节，以后又读2个完整字段。

⑥ 存取顺序处理文件的程序格式小结

a) 把资料存入顺序处理文件的程序格式：

```
10  D$=CHR$(4):REM CTRL-D
20  PRINT D$; "OPEN 文件名"
30  PRINT D$; "DELETE 文件名"
```

```
40  PRINT D$; "OPEN 文件名"
50  PRINT D$; "WRITE 文件名"
60  PRINT 资料
70  PRINT D$; "CLOSE"
```

b）从顺序处理文件中取资料的程序格式：

```
10  D$=CHR$(4):REM CTRL-D
20  PRINT D$; "OPEN 文件名"
30  PRINT D$; "READ 文件名"
40  INPUT 资料
50  PRINT D$; "CLOSE"
```

c）在顺序处理文件中增加资料的程序格式：

```
10  D$=CHR$(4):REM CTRL-D
20  PRINT D$; "APPEND 文件名"
30  PRINT D$; "WRITE 文件名"
40  PRINT 资料
50  PRINT D$; "CLOSE"
```

d）向顺序处理文件的任意字段中重写入资料的程序格式：

```
10  D$=CHR$(4):REM CTRL-D
20  PRINT D$; "OPEN 文件名"
30  PRINT D$; "POSITION 文件名, Rp"
40  PRINT D$; "WRITE 文件名"
50  PRINT 资料
60  PRINT D$; "CLOSE"
```

程序中的Rp指的是相对字段位置。

e）从顺序处理文件的任意字段开始读资料的程序格式：

```
10  D$=CHR$(4):REM CTRL-D
```

```
20  PRINT D$; "OPEN 文件名"
30  PRINT D$; "POSITION.文件名,Rp"
40  PRINT D$; "READ 文件名"
50  INPUT 资料
60  PRINT D$; "CLOSE"
```

f）直接从指定字节开始写资料的格式

```
10  D$=CHR$(4): REM CTRL-D
20  PRINT D$; "OPEN 文件名"
30  PRINT D$; "WRITE 文件名, Bb"
40  PRINT 资料
50  PRINT D$; "CLOSE"
```

程序中的Bb指的是绝对字节位置。

g）直接从指定字节开始读资料的格式

```
10  D$=CHR$(4): REM CTRL-D
20  PRINT D$; "OPEN 文件名"
30  PRINT D$; "READ 文件名, Bb"
40  INPUT 资料
50  PRINT D$; "CLOSE"
```

2.11.9 随机存取文件

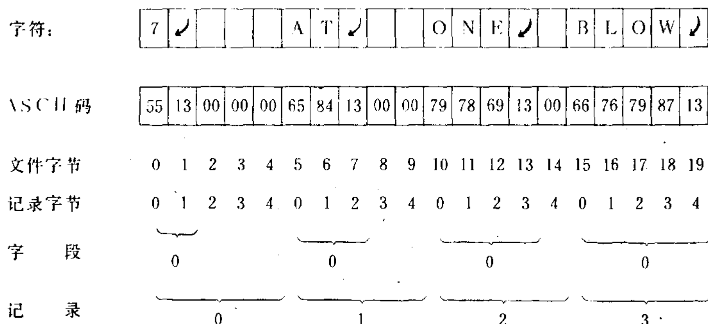
顺序处理文件虽然可以指定字段和字节存取文件，但是必须知道字节确切位置，存取麻烦，容易出错。随机存取文件就具有更大的灵活性。

随机存取文件存贮格式

随机存取文件由许多记录项组成，记录编号从零开始，每个记录项事先已指定可以存贮的字符个数，每个记录能存贮的字符个数称为记录的长度。根据实际问题的需要，显然有的记录充满了字符，有的记录没有充满。要求一个随机存

取文件内的记录长度相同的目的是便于从文件任何部分很快找到信息。随机存取文件存贮资料格式如下图所示：

随机存取文本文件格式个例图



字符:	7	↲				A	T	↲			O	N	E	↲		B	L	O	W	↲
ASCII码	55	13	00	00	00	65	84	13	00	00	79	78	69	13	00	66	76	79	87	13
文件字节	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
记录字节	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
字段	0					0					0					0				
记录	0					1					2					3				

以上图例中有 4 个记录，每个记录有 1 个字段，每个字段的字节数不同，因此有的记录已充满字节，多数记录并没充满字节。

字符串随机文件的存取程序

① 下面的例子在一个指定记录内存入几个字符串：

```
10  REM MAKE MAILER
20 D$ =  CHR$ (4): REM CTRL-D
30  INPUT "NAME=";N$
40  INPUT "PHONE=";P$
50  INPUT "ZIP CODE=";Z$
60  PRINT D$;"OPEN MAILER,L200"
70  PRINT D$;"WRITE MAILER,R1"
80  PRINT N$: PRINT P$: PRINT Z$
70  PRINT D$;"CLOSE MAILER"
100  END
```

第20行赋值一个控制字；第30行到50行键盘输入字符串；第60行打开随机文件 MAILER， L是记录字节参数，指定每个记录最多容纳的字节长度为 200 个字符；第70行命令在 MAILER 文件的记录 R1 中写入资料；第 80 行是三个写入资料的语句；最后第90行关闭这个随机文件。与建立顺序文件相比较，只是“打开”和“写入”语句的格式不同，其它语句类似。

运行这个程序就能在磁盘上建立一个T型文件（文本文件）即一个随机存取的文件，通过监控可显示建立文件的情况如下：

```
]RUN
NAME=A
PHONE=02312
ZIP CODE=022
OPEN MAILER,L200
WRITE MAILER,R1
A
02312
022
CLOSE MAILER
```

以下这个程序可以从上面建立的随机文件的记录 1 中取出资料，其运行情况也同时列出：

```
10  REM RETRIEVE MAILER
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN MAILER,L200"
40  PRINT D$;"READ MAILER,R1"
50  INPUT N$,P$,Z$
60  PRINT D$;"CLOSE MAILER"

]MON C,I,O
```

```
]RUN
OPEN MAILER,L200
READ MAILER,R1
?A
??02312
??022
CLOSE MAILER
```

程序第20行置控制字；第30行打开该随机文件，指定字节长度必须与原来建文件时指定的一样，不然就会出错；第40行执行读数命令，指定第一个记录（因为资料是存在第一个记录内的）；第50行从随机文件读取 3 个字符串；第60行关闭文件。监控运行把读数情况都显示出来，可以看到凡是 INPUT 语句，也带有问号标志，然后自动从随机文件中读入字符串。

第50行也可以写为 3 个程序行，运行的结果也能读出 3 个字符串：

```
50  INPUT  N$
52  INPUT  P$
54  INPUT  Z$
```

② 在几个记录中存取资料

下面的程序在指定的几个记录中存入一批资料：

```
5   REM MAKE FILE
10   INPUT "FILE NAME=";A$
20 D$ =  CHR$ (4): REM CTRL-D
30   PRINT D$;"OPEN";A$
40   PRINT D$;"DELETE";A$
50   PRINT D$;"OPEN";A$;",L30"
60   FOR I = 12 TO 15
70   PRINT D$;"WRITE";A$;",R";I
80   PRINT "CODE=";I
90   NEXT I
```

```
100   PRINT D$;"WRITE";A$;",R13"
110   PRINT "DOS"
120   PRINT D$;"CLOSE";A$
130   END
```

第10行键盘输入所建文件名；第20行置控制字；第30行打开文件；第40行删除这个文件，这两行所起作用与以前对顺序处理文件的处理一样，即避免新建文件与原先磁盘上可能存在的同名旧文件资料混淆而采取的措施；第50行再打开这个文件，指定记录长度是30个字节；第60行进入循环；第70行执行写入命令，每写入一个记录都须执行一次写命令；第80行写入文件中的资料；第90行循环终端语句；第 100 行执行在第13个记录中写入字符串的命令；第 110 行为写入资料；第120行关闭文件。

必须注意的几点：

a）打开文件、删除、再打开的格式虽然可以避免新旧文件资料的混淆，但是若是要保持原有文件中各记录的资料不变，要在别的记录写入资料时，用这样的格式就不行了，因为用了这样的格式把原文件中有用的资料也给删除了，因此这时写入资料的格式只能用打开文件，指定记录，直接写入的格式；

b）指定记录长度必须合适。太小不够用，就会存错位置，太大则浪费存贮空间；

c）每个记录写入资料之前都必须先发出一个写的命令；

d）最后一定要关闭文件。

监控运行这个程序的情况如下：

```
]RUN
FILE NAME=AAA
```

```
OPENAAA
DELETEAAA
OPENAAA,L30
WRITEAAA,R12
CODE=12
WRITEAAA,R13
CODE=13
WRITEAAA,R14
CODE=14
WRITEAAA,R15
CODE=15
WRITEAAA,R13
DOS
CLOSEAAA
```

下面程序可从以上建立的文件中读出字符串。

```
10  REM RETRIEVE FILE
20 D$ =  CHR$ (4): REM CTRL-D
30  INPUT "FILE NAME=";B$
40  PRINT D$;"OPEN";B$;",L30"
50  FOR J = 12 TO 15
60  PRINT D$;"READ";B$;",R";J
70  INPUT C$
80  IF  LEFT$ (C$,3) = "DOS" THEN
      PRINT "RECORD ";J;" WAS CHA
     NGED"
90  NEXT J
100  PRINT D$;"CLOSE";B$
110  END
```

第40行打开文件，指定记录长度为30；第60行读命令位于循环语句之内，表明每读取一个记录的内容必须先执行一次读命令，并指定记录号；第70行执行从数据文件中读取资料；

第80行是比较检索，假如在这个记录中字符串的头三个字符是DOS，则打印出一个信息；第100行关闭文件，这个程序运行情况如下：

```
ÜMON C,I,O

ÜRUN
FILE NAME=AAA
OPENAAA,L30
READAAA,R12
?CODE=12
READAAA,R13
?DOS
RECORD 13 WAS CHANGED
READAAA,R14
?CODE=14
READAAA,R15
?CODE=15
CLOSEAAA
```

存取数据随机文件的程序

① 单个记录的读写

存取数据随机文件程序格式与存取字符随机文件程序是一样的，下面是一个简单的例子，建立的数据文件中指定记录长度是200个字节，为了以后醒目起见，把这个字节长度列为数据文件名字的一部分，因而这个数据文件的名字为QAZ:L200，指定资料存在第五个记录之中：

```
10  REM MAKE QAZ:L200
20  DIM A(12)
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN QAZ:L200"
```

```
50   PRINT D$;"DELETE QAZ:L200"
60   PRINT D$;"OPEN QAZ:L200,L200"

70   PRINT D$;"WRITE QAZ:L200,R5"
80   FOR I = 1 TO 12
90 A(I) = I * I
100   PRINT A(I)
110   NEXT
120   PRINT D$;"CLOSE QAZ:L200"
```

监控运行结果是:

```
]MON C,I,O

]RUN
OPEN QAZ:L200
DELETE QAZ:L200
OPEN QAZ:L200,L200
WRITE QAZ:L200,R5
1
4
9
16
25
36
49
64
81
100
121
144
CLOSE QAZ:L200
```

从这个随机文件中取资料的程序和运行情况如下:

```
10   REM RETRIEVE QAZ:L200
20   DIM A(12)
```

```
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN QAZ:L200,L200"

50  PRINT D$;"READ QAZ:L200,R5"
60  FOR I = 1 TO 12
70  INPUT A(I)
80  NEXT
90  PRINT D$;"CLOSE QAZ:L200"

]MON C,I,O

]RUN
OPEN QAZ:L200,L200
READ QAZ:L200,R5
?1
?4
?9
?16
?25
?36
?49
?64
?81
?100
?121
?144
CLOSE QAZ:L200
```

② 多个记录的读写

下面的例子表示在几个记录中各自存一批数据。数据文件名字是EX202:L50，指定的每个记录长度是50个字节：

```
10  REM MAKE EX202:L50
20  DIM A(5,15)
30  FOR I = 3 TO 5
40  FOR J = 1 TO 15
```

```
50   READ A(I,J)
60   NEXT
70   NEXT
80 D$ =  CHR$ (4): REM CTRL-D
90   PRINT D$;"OPEN EX202:L50,L50,
      D2"
100   FOR I = 3 TO 5
110   PRINT D$;"WRITE EX202:L50,R"
      ;I
120   FOR J = 1 TO 15
130   PRINT A(I,J)
140   NEXT
150   NEXT
160   PRINT D$;"CLOSE"
300   DATA 1,2,3,4,5,6,7,8,9,10,11
      ,12,13,14,15
310   DATA 2,4,6,8,10,12,14,16,18,
      20,22,24,26,28,30
320   DATA 3,6,9,12,15,18,21,24,27
      ,30,33,36,39,42,45
```

运行的结果，在第三到第五个记录，每个记录存贮15个数据，第 110 行写的命令在循环语句之内，表明在每个记录中写数据之前，先要发出一个写的命令。

下面这个程序可从这个数据文件中取出数据：

```
10   REM RETRIEVE EX202:L50
20   DIM B(5,15)
30 D$ =  CHR$ (4): REM CTRL-D
40   PRINT D$;"OPEN EX202:L50,L50"

50   FOR I = 3 TO 5
60   PRINT D$;"READ EX202:L50,R";I

70   FOR J = 1 TO 15
80   INPUT B(I,J)
```

```
90  NEXT
100  NEXT
110  PRINT D$;"CLOSE EX202:L50"
```

要注意的是，在取数据文件时，指定记录长度必须与建文件时指定的一样；READ 命令必须在每次从每个记录读入数据之前发出。

③ 指定记录长度太小出现的问题

建随机数据文件时，就有一个指定每个记录所能容纳的最多字节容量问题，太大浪费磁盘空间，太小存取数据会出错。下面的例子表示，指定的记录长度太小，存取资料发生重叠的情形。建立数据文件的程序如下：

```
10  REM MAKE EX202:L15
15  DIM A(5,15)
20  FOR I = 3 TO 5
30  FOR J = 1 TO 15
40  READ A(I,J)
50  NEXT
60  NEXT
80 D$ =  CHR$ (4): REM CTRL-D
90  PRINT D$;"OPEN EX202:L15"
100  PRINT D$;"DELETE EX202:L15"
110  PRINT D$;"OPEN EX202:L15,L15
     "
120  FOR I = 3 TO 5
130  PRINT D$;"WRITE EX202:L15,R"
     ;I
140  FOR J = 1 TO 15
150  PRINT A(I,J)
160  NEXT
170  NEXT
180  PRINT D$;"CLOSE EX202:L15"
200  DATA 1,2,3,4,5,6,7,8,9,10,11
```

```
        ,12,13,14,15
210     DATA 2,4,6,8,10,12,14,16,18,
        20,22,24,26,28,30
220     DATA 3,6,9,12,15,18,21,24,27
        ,30,33,36,39,42,45
```

第110行指定记录长度为15,无法容纳实际往记录中存入15个数据的长度，监控运行情况似乎正常，实际上数据存放位置已混乱。

从这个文件中取数据的程序以及监控运行的情况就可以证明这一点:

```
10   REM RETRIEVE EX202:L15
20   DIM B(5,15)
30 D$ =  CHR$ (4): REM CTRL-D
40   PRINT D$;"OPEN EX202:L15,L15"

50   FOR I = 3 TO 5
60   PRINT D$;"READ EX202:L15,R";I

70   FOR J = 1 TO 15
80   INPUT B(I,J)
90   NEXT
100   NEXT
110   PRINT D$;"CLOSE EX202:L15"


]RUN
OPEN EX202:L15,L15
READ EX202:L15,R3
?1
?2
?3
?4
?5
```

```
?6
?7
?82
?4
?6
?8
?10
?12
?13
?6
READ EX202:L15,R4
?2
?4
?6
?8
?10
?12
?13
?6
?9
?12
?15
?18
?21
?24
?27
READ EX202:L15,R5
?3
?6
?9
?12
?15
?18
?21
?24
?27
?30
```

```
?33
?36
?39
?42
?45
CLOSE EX202:L15
```

可见取到的只是混乱的结果。

④ 指定记录、字节读写文件

在随机存取文件中，不但可以读写整个指定记录，而且可以在指定记录中，从指定的字节开始读写资料。下面这个例子表示从上面已建立了的随机文件EX 202 ：L 50的记录 3 的编号为第六个字节起取数据（字节编号是从第零号开始的，标点符号和回车也是字节），由于在这个记录中原先只存了15个数据，现在是从第六个字节开始读数据，故在这个记录中最多只能读到12个数据。第50行指明了从EX 202 的第三个记录的第六个字节开始读数据。程序及运行情况如下：

```
10  REM RETRIEVE EX202:L50-2
20  DIM B(15)
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN EX202:L50,L50"

50  PRINT D$;"READ EX202:L50,R3,B
     6"
60  FOR J = 1 TO 12
70  INPUT B(J)
80  NEXT
90  PRINT D$;"CLOSE EX202:L50"

]MON C,I,O

]RUN
```

```
OPEN EX202:L50,L50
READ EX202:L50,R3,B6
?4
?5
?6
?7
?8
?9
?10
?11
?12
?13
?14
?15
CLOSE EX202:L50
```

在每个记录中，字节编号从零开始，EX 202 :L 50文件的第三个记录的第六个字节正好是 4 。

随机存取文件小结

① 存资料格式：

```
10  D$ =CHR$ ( 4 ):REM CTRL-D
20  PRINT D$；"OPEN 文件名，Lj，Dd"
30  PRINT D$；"WRITE 文件名，Rr"
40  PRINT 资料
50  PRINT D$；"CLOSE"
```

取资料格式：

```
10  D$ =CHR$ ( 4 ):REM CTRL-D
20  PRINT D$；"OPEN 文件名，Lj，Dd"
30  PRINT D$；"READ 文件名，Rr"
40  INPUT 资料
50  PRINT D$；"CLOSE"
```

其中Lj是记录长度，Dd是指定的驱动器号，Rr是记录号。要求j＝1—32767，包括字符，回车符号等。

② 记录长度设计要合适，太大浪费磁盘空间，太小则存取混乱。记住在同一随机文件中的每个记录长度必须相同，设计时应该取所有记录中计划要存贮的最长字节数作为指定的记录长度。在设计取数据文件的程序时，指定的记录长度应该与建立这个数据文件时指定的记录长度相同。

③ 读写每个记录时，必须每次发出READ或WRITE命令。

④ 设计取随机文件的程序之前，先要清楚知道存贮的随机文件的具体格式，特别是在指定字节存取时更是这样，所以对于已建的随机文件应该有档案记载，以便以后查阅。

2.11.10 程序的链接运行

有时为了使几个程序通过磁盘能连贯自动运行或为了节省内存把一个大的程序分成几段分别存入磁盘，运行时又可连续运行，这就需要使用程序链接运行技术。主要有两种方法：

简单的链接运行

一个程序运行结束后若要求自动调进和运行另一个已在磁盘上的程序，最简单的办法是在这个程序的末尾加一个RUN的DOS命令。例如在磁盘上已存有如下程序（文件名LINK 2）：

```
10  REM PROGRAM LINK 2
20  DIM A(11),B(11)
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN EX103"
50  PRINT D$;"READ EX103"
```

```
60  FOR I = 1 TO 11
70  INPUT A(I)
80 B(I) = A(I) * I
90  NEXT
95  PRINT D$;"CLOSE"
100  PRINT D$;"PR#1"
110  FOR I = 1 TO 11
120  PRINT A(I),B(I)
130  NEXT
140  PRINT D$;"PR#0"
```

```
]RUN
MEAN X = 6          S = 1.22          C.V. = .2
-.82  0  1.63  .82  .82  -1.63  0  -.82  0

MEAN X = .5         S = .21           C.V. = .42
1.41  -.94  .47  .94  0  -.47  .94  -1.41  -.94
```

现在主机中建立了如下程序:

```
10  REM PROGRAM LINK 1
20  DIM A(11)
30 D$ =  CHR$ (4): REM CTRL-D
40  PRINT D$;"OPEN EX103"
50  PRINT D$;"DELETE EX103"
60  PRINT D$;"OPEN EX103"
70  PRINT D$;"WRITE EX103"
80  FOR I = 1 TO 11
90  READ A(I)
100  PRINT A(I)
110  NEXT
120  PRINT D$;"CLOSE"
130  PRINT D$;"RUN LINK 2"
300  DATA 10,20,30,40,50,60,70,80
     ,90,100,110
```

程序40—70行建立一个数据文件的DOS命令，数据文件EX103中存有11个数据；第130行是链接运行磁盘上LINK 2文件的DOS命令。只要运行现在主机中的这个程序，就会在磁盘上建立一个T型文件EX103,运行终了立即自动调入磁盘上的文件LINK 2,并运行。即先读进T文件EX103上的数据，然后开动打印机（LINK2程序的第100行），打出11对A(I)，B(I)的值，最后关闭打印机(第140行)，运行的结果是：

10	10
20	40
30	90
40	160
50	250
60	360
70	490
80	640
90	810
100	1000
110	1210

这种程序链接运行的特点，是主机中的程序只是发出了调入和运行指定的磁盘文件的命令。除此之外，这两个程序可以没有丝毫关系，运行完毕，主机中只有后来调入的程序，原先主机中程序已被清除。

内存相关的链接运行（或链接运行）

在这种设计下，运行主机中的程序即可自动调入并运行磁盘上指定的程序或文件，而主机程序中的变量值仍保存下来，在调入的程序中继续发挥作用。由于链接运行中用到了

应用程序CHAIN，所以在做链接运行之前，需先把系统盘上的CHAIN程序转移到现在这个磁盘上（利用FID程序转移，请参考2.11.12）

链接运行方式常常应用于当源程序比较长，一次装不进内存时，把这个程序分割成几个较短的程序，按照链接运行设计，通过磁盘使整个程序仍能自动连续运行，中间不必有人工干预。以下举一个简单的例子，说明链接运行的用法。

假设有如下程序：

```
10  REM CHA
20  DIM A(15)
30  FOR I = 1 TO 15
40 A(I) = I / 2
50  PRINT A(I),
60  NEXT
70  PRINT
75 D$ =  CHR$ (4): REM CTRL-D
80  PRINT D$;"PR#1"
90  FOR I = 1 TO 10
100 B(I) = I * I
110  PRINT B(I),
120  NEXT
130  PRINT
140  FOR I = 1 TO 15
150  PRINT A(I) * I,
160  NEXT
170  PRINT
180  PRINT D$;"PR#0"
```

假设认为这个程序太长，内存装不下，则可以分割成两个较短的程序CHA-1和CHA-2，先把CHA-2存入磁盘，然后把CHA-1键入主机，通过内存相关的链接运行语句，只要运行CHA-1，就会自动链接运行磁盘上的CHA-2程

序，完成全部运行任务。设计的CHA-1程序如下：

```
10  REM CHA-1
20  DIM A(15)
30  FOR I = 1 TO 15
35 A(I) = I / 2
40  PRINT A(I),
50  NEXT
60  PRINT
100  PRINT  CHR$ (4);"BLOAD CHAIN
     ,A520"
110  CALL 520"CHA-2"
```

其中程序第100行和110行是链接运行的语句，CHA-2则为磁盘上已存在的文件，注意第110行520后面必须紧跟引号。

磁盘上预先已存入的CHA-2程序如下：

```
10  REM CHA-2
15 D$ =  CHR$ (4): REM CTRL-D
17  PRINT D$;"PR#1"
20  FOR I = 1 TO 10
30 B(I) = I * I
40  PRINT B(I),
50  NEXT
55  PRINT
60  FOR I = 1 TO 15
70  PRINT A(I) * I,
80  NEXT
90  PRINT
100  PRINT D$;"PR#0"
```

这时运行主机上的程序，即可得到如下结果：

```
]RUN
.5              1               1.5
2               2.5             3
3.5             4               4.5
5               5.5             6
6.5             7               7.5

1               4               9
16              25              36
49              64              81
100
.5              2               4.5
8               12.5            18
24.5            32              40.5
50              60.5            72
84.5            98              112.5
```

显然这是主机上的程序与磁盘上程序CHA-2运行后共同的结果。

2.11.11 程序中编写入DOS命令时要注意的细节

在程序中使用DOS命令能使主机与外部设备方便地交换信息，但一定要严格按照编写格式使用这些命令才行，即使标点符号对于DOS命令操作的影响也是够大的。例如下面这个程序及其运行结果：

```
10 D$ =  CHR$ (4): REM CTRL-D
20  FOR I = 1 TO 10
30  PRINT I,
40  NEXT
50  PRINT D$;"RUN LINK 2"

]RUN
1               2               3
4               5               6
```

```
7                    8                    9
10                   RUN LINK 2
```

第30行规定了按标准格式输出，逗号起到了预留显示空白位置的作用，结果影响到第50行的连接运行命令，使它也变成了打印输出的对象，不能实现连接运行的设计。若是在程序中加一行 45 PRINT 就可结束标准输出格式，顺利执行连接运行的命令。

2.11.12 FID程序

系统主盘上有一个二进制类型的程序文件，称为FID，为管理磁盘文件提供了方便。由于它是二进制类型的文件，从磁盘调入并运行的命令是BRUN。键入BRUN FID命令后将在屏幕上显示：

```
****************************************
*       APPLE ][ FILE DEVELOPER        *
*                                      *
*           FID VERSION M              *
*                                      *
* COPYRIGHT 1979 APPLE COMPUTER INC.   *
****************************************

CHOOSE ONE OF THE FOLLOWING OPTIONS


       <1>   COPY FILES
       <2>   CATALOG
       <3>   SPACE ON DISK
       <4>   UNLOCK FILES
       <5>   LOCK FILES
       <6>   DELETE FILES
       <7>   RESET SLOT & DRIVE
       <8>   VERIFY FILES
```

<9>　QUIT

WHICH WOULD YOU LIKE?

屏幕提示 9 种功能供选择，只要键入指定的数字就进入操作，9 种功能是：

COPY FILES

用于整个源盘和单个文件复制，它与主盘上另一个 COPYA 文件的区别是：

① COPY FILES 程序，既可用于整片“源盘”的复制，也可对单个文件复制；COPYA 程序只能对整片“源盘”的复制；

② COPY FILES 程序可以用来复制单个数据文件；COPYA 程序则无能为力；

③ 用 COPY FILES 程序复制时，复制盘必须预先做格式化工作；COPYA 程序用于复制磁盘时，复制盘不需要预先格式化，因为复制中，它已自动做格式化工作了。

COPY EILES 程序的用法与 COPYA 程序类似，但显示要求输入文件名时，有如下几种处理方法备选：

a）若键入某文件名，表示只复制具有该文件名的文件；

b）若键入＝号，表示复制整个“源盘”；

c）若键入＝EF，表示对“源盘”中所有以 EF 结尾的文件复制；

d）若键入 NS＝，则表示复制“源盘”中所有以 NS 开头的文件；

e）若键入＝AB＝，表示对“源盘”中所有包含 AB 的文件进行复制。

CATALOG

列出磁盘文件目录。

SPACE ON DISK

显示磁盘已占用多少扇区，还有多少扇区可用。

UNLOCK FILES

对指定文件开锁。

LOCK FILES

对指定文件加锁。

DELETE FILES

删除指定文件。

RESET SLOT & DRIVE

取消当前省略说明选择的槽口和驱动器号。

VERIFY FILES

检查指定文件占用的扇区。

—QUIT

退出FID控制

2.11.13 MAXFILES命令

DOS允许最多同时使用16个文件，MAXFILES命令确定可以使用多少个文件。主机引入DOS时，系统已自动分配三个磁盘文件缓冲区，规定同时只能打开3个文件，而使用MAXFILES语句就使用户有选择更多文件缓冲区的手段。下面是MAXFILES命令用在程序中的例子及其运行情况：

```
10  PRINT  CHR$ (4);"MAXFILES 1"
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN EX101"
40  PRINT D$;"READ EX101"
```

```
50   INPUT A
60   PRINT D$;"OPEN EX102"
70   PRINT D$;"READ EX102"
80   INPUT B
100  PRINT D$;"CLOSE"

]MON C,I,O

]RUN
MAXFILES 1
OPEN EX101
READ EX101
?1
OPEN EX102

NO BUFFERS AVAILABLE

BREAK IN 60
```

其中第10行规定只能打开1个文件，程序设计中却要求打开2个文件，所以运行出错，执行到60行发生"缓冲区不够用"的错误。

事实上，对于每个指定的文件，MAXFILES 命令指定留出 595 个字节的存贮空间作为文件缓冲区，指定的文件数少于要求打开的文件数，当然就出错。使用MAXFILES命令应注意：

(1) MAXFILES 命令可用作直接命令，格式是：

MAXFILES n

其中 n是指定文件数，但n=1—16。

MAXFILES 用在程序中必须与控制字一起使用。

(2) 打开文件以后又CLOSE，则表示已退出这个文件缓冲区，所以若上例程序中增加程序行 55 PRINT D$;

“CLOSE”,则运行不会出错。

(3) MAXFILES 用作直接命令时，必须在引入程序之前执行，否则有可能破坏数据信息，造成运行混乱；MAXFILES 用于程序中时，也必须放在程序的首行，避免运行混乱。

(4) 由于使用MAXFILES 命令指定可以打开的文件个数，实质上也就是要求预留多少缓冲区，预留多少内存，所以应在设计程序时，考虑各方面对内存要求的平衡。使用MAXFILES 命令要求同时打开的文件数量时，应充分注意。

2.11.14 EXEC命令

DOS 的EXEC 命令是运行的命令，但与RUN 命令不同之处在于：EXEC 命令运行的已在磁盘上的顺序文本文件，这个文件必须由BASIC 命令和语句组成。也就是运行EXEC命令时，APPLE Ⅱ 的控制权转入文件内。

(1) EXEC 命令能快速检查数据文件的内容，用法是：

① 全部检查所用格式：

EXEC 数据文件名

要全部检查数据文件的内容，应该先发出磁盘监控命令(MON C, I, O)，然后发出EXEC 命令。例如我们曾在磁盘上建立过一个数据文件EX102，要检查它的内容，可按如下方式操作：

```
]MON C,I,O
]EXEC EX102
]50
```

```
ü60

ü70

ü80

ü90

ü100

ü110

ü120

ü130

ü140
```

② 从指定字段开始检查数据文件，格式是：

EXEC 文件名，Rp

其中Rp指从文件的第p+1个字段开始执行命令，p=0表示从文件的第1个字段开始，p=1表示从文件的第2个字段开始执行命令，等等。当然为了屏幕上能看到显示，键入以上命令之前也须先执行MON C，I，O命令。以下就是对磁盘数据文件EX102指定字段执行EXEC命令的情况：

```
üEXEC EX102,R4

ü90

ü100

ü110
```

```
0120

0130

0140
```

(2) EXEC 命令可以从磁盘调进一个包含有各种命令的文件，并自动执行。当然在这之前必须在磁盘上建立这样的一个文件。下面的例子表示先在磁盘上建立了一个文件 QQPP，其中包括三个 CATALOG 命令，一个INT 命令。建立这个顺序处理文件QQPP的程序如下：

```
10  REM MAKE T FILE
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN QQPP"
40  PRINT D$;"DELETE QQPP"
50  PRINT D$;"OPEN QQPP"
60  PRINT D$;"WRITE QQPP"
70  FOR I = 1 TO 3
80  PRINT "CATALOG"
90  NEXT
100  PRINT "INT"
110  PRINT D$;"CLOSE"
```

再对文件QQPP 执行EXEC 命令，就会看到把当时磁盘上的目录列出三次，最后并转变为整型BASIC 的提示符。

(3) EXEC 命令可以从磁盘上调进一个包含有若干程序行的文件，接在主机中已有的文件之中，即通过磁盘把程序连接起来。当然在这之前先要把程序行在磁盘上建立成文本文件形式。下面这个例子表示先把一些程序行在磁盘上建立为文件形式：

```
10 D$ =  CHR$ (4): REM CTRL-D
20  PRINT D$;"OPEN QP"
30  PRINT D$;"DELETE QP"
40  PRINT D$;"OPEN QP"
50  PRINT D$;"WRITE QP"
60  LIST 100,160
70  PRINT D$;"CLOSE"
100  PRINT "CATALOG"
110  PRINT "INT"
120  PRINT "FP"
130  FOR I = 1 TO 3
140  PRINT "CATALOG"
150  NEXT I
160  PRINT "INT"
```

程序中通过20—70行把程序行100—160行以顺序处理文件形式存入磁盘，文件名QP，程序运行情况的监控记录是：

```
]RUN
OPEN QP
DELETE QP
OPEN QP
WRITE QP

100  PRINT "CATALOG"
110  PRINT "INT"
120  PRINT "FP"
130  FOR I = 1 TO 3
140  PRINT "CATALOG"
150  NEXT I
160  PRINT "INT"
CLOSE
CATALOG
INT
FP
CATALOG
CATALOG
```

```
CATALOG
INT
```

对文件QP执行EXEC命令运行情况是:

```
]EXEC QP

]

]100  PRINT "CATALOG"

]110  PRINT "INT"

]120  PRINT "FP"

]130  FOR I = 1 TO 3

]140  PRINT "CATALOG"

]150  NEXT I

]160  PRINT "INT"
```

结果把100—160行程序引入主机，假如当时主机中还有别的程序，这段程序就插到原有程序中去了。

§ 2.12 DOS的错误信息

当DOS检测出磁盘系统方面的错误时，会发出"嘟"的一声，显示错误类型和内容，程序停止执行。可以根据如下方法区别错误的类型:

(1) 我们在§ 2.10介绍过的APPLESOFT 错误信息前面有一问号，例如? SYNTAX ERROR (语法错);

(2) 整型BASIC错误信息前面有三个星号，例如＊＊

*SYNTAX ERR（语法错）；

(3) DOS 错误信息前面没有符号，例如 SYNTAX ERROR（语法错）。

以下是DOS 出错信息内容，PEEK (222) = A，表示在内存222单元处该DOS 错误信息的编码。

DISK FULL

磁盘已经满了，不能再存信息，REEK(222) = 9。

END OF DATA

文件中的数据不够用，PEEK(222) = 5。

FILE LOCKED

文件锁住了，不能再对这个文件进行保存、换名或删除，PEEK(222) = 10。

FILE NOT FOUND

磁盘中没有指定的文件，PEEK(222) = 6

FILE TYPE MISMATCH

所用命令与文件类型不匹配，例如 LOAD，RUN和SAVE命令等只能用于APPLESOFT 或整型BASIC程序文件；OPEN, READ, WRITE, APPEND, EXEC 等命令只能用于文本文件，BLOAD, BSAVE与BRUN只能用于二进制文件。PEEK(222) = 13

I/O ERROR

无法在磁盘上存取信息，原因可能是没有放好磁盘；可能磁盘片有损伤或驱动器故障；有时在一个驱动器上存的信息在别的驱动器上读不出来，可在原先驱动器上试试等等。

LANGUAGE NOT AVAILABLE

内存或磁盘中没有所用命令所要求的语言，PEEK(222) = 1。

NO BUFFERS AVAILABLE

在内存中所规定的（用MAXFILES 命令）文件缓冲区不够用。PEEK(222)=12。

NOT DIRECT COMMAND

下列命令不允许在直接命令方式中使用，例如APPEND，OPEN，READ，WRITE，POSITION等只能用在程序方式之中。PEEK(222)=15。

PROGRAM TOO LARGE

调入主机内存的程序太大，内存不足。PEEK(222)=14。

RANGE ERROR

在DOS命令中某个参数的值超过了指定的范围。PEEK(222)=2或3。下面的表摘要列出DOS命令中某几种参数的允许范围：

参	数	字母	范围 最小值	范围 最大值
顺序处理文件	字节	B	0	32767
	相对字段	R	0	32767
	绝对字段(EXEC)	R	0	32767
随机存取文件	记录长度	L	1	32767
	记录号	R	0	32767
MAXFILES	n	n	1	16

SYNTAX ERROR

语法错，指DOS命令中拼字拼错，标点符号用错等。PEEK(222)=11。

VOLUME MISMATCH

DOS命令中指定磁盘卷号与已存入磁盘上的卷号不符。PEEK(222)=7。

WRITED PROTECTED

磁盘已进行了写入保护，不许写入。PEEK(22)=4。

§2.13 图形显示语句

在程序中利用下列命令或语句，就可以控制图形显示，达到所要求的输出形式。

2.13.1 低分辨图形显示（最大为40行×48列的图形显示区）

GR

GR为进入低分辨图形显示的命令或语句，它有两个显示区域供选择，分别称为显示的第一页和第二页。

① 执行GR语句后，屏幕显示即进入第一页，屏幕清成全黑，区域为40×40点阵，屏幕底部留出4行正文显示窗口；

② 执行GR语句后再键入POKE-16302，0语句，整个屏幕全黑，区域为40×48点阵。

COLOR

COLOR=n语句规定了低分辨率作图的颜色，0≤n≤255，代表了16种颜色中的一种，当n>16时，可选择的颜色数按16取模。

n值代表的颜色如下：

0	黑色	4	深绿	8	褐色	12	绿色
1	品红	5	灰色	9	橙色	13	黄色
2	深蓝	6	天蓝	10	灰色	14	湖蓝
3	紫色	7	淡蓝	11	粉红	15	白色

由于系统执行了GR语句后，自动地使COLOR=0，

所以执行绘图语句前一定先给COLOR赋值。

PLOT

格式：PLOT X，Y

X，Y是算术表达式，PLOT X, Y表示在屏幕坐标（X，Y）上画一个点。要求 $0 \leqslant X \leqslant 39$，$0 \leqslant Y \leqslant 47$，超过此两限度时，将出现出错信息。

HLIN

画水平线语句格式：

HLIN X_1，X_2 AT Y

其中 X_1，X_2 和 Y都是算术表达式，以上格式表示从屏幕（X_1，Y）点画一条水平线到（X_2，Y）点。

VLIN

画垂直线语句格式：

VLIN Y_1，Y_2 AT X

其中 Y_1，Y_2 和 X都是算术表达式，以上格式表示从屏幕坐标（X，Y_1）点画一条垂线到（X, Y_2）点。要求：$0 \leqslant X \leqslant 39$，$0 \leqslant Y \leqslant 47$。

SCRN（X，Y）

这是一个函数，X和Y是算术表达式，这个函数求出屏幕坐标（X，Y）上颜色的数码。要求 $0 \leqslant X \leqslant 39$，$0 \leqslant Y \leqslant 47$。

低分辨图形的例子

```
10  GR
20  COLOR= 15
30  PLOT 12,12
40  HLIN 1,39 AT 20
50  VLIN 1,47 AT 20
```

这个程序运行的结果在屏幕上的12行和12列处显示一个点，20行处一条水平线，20列处一条垂直线。

2.13.2 高分辨图形显示（最大为280行×192列的图形显示区）

HGR 和HGR 2

① HGR

命令或语句HGR 进入高分辨图形第一页，图形显示区域为280×160点阵，屏幕底端留出四行作为正文显示区域，这种显示需拥有内存容量为16K以上的系统，以便使APPLESOFT 能留驻内存不被破坏。

执行HGR，并键入POKE-16302，0，则进入全屏幕显示，显示区域为280×192点阵。

② HGR 2

命令或语句HGR 2 进入高分辨图形第 2 页，显示区域为280×192点阵，为全屏幕图形显示；执行HGR 2 后，键入POKE-16301，0，则进入280×160点阵的图形显示区，屏幕底端留出 4 行正文显示区域。这种显示需拥有内存容量为24K以上系统，以便APPLESOFT 能同时留存内存。若想同时再使用DOS，则至少要拥有内存36K以上。

HCOLOR

规定使用颜色的语句格式是HCOLOR =n，0≤n≤7，但实际上只有 6 种颜色可供选择。编号为：

0 黑色 1 绿色 2 蓝色 3 白色

4 黑色 5 紫色 6 红色 7 白色

HPLOT

有几种用法：

① HPLOT X，Y

在屏幕坐标（X，Y）上画一个点；

② HPLOT X_1，Y_1 TO X_2，Y_2

从屏幕坐标(X_1，Y_1）开始画直线到坐标（X_2，Y_2）。例如可从右上角到左下角划出一条对角线；

③ HPLOT TO X_2，Y_2

从上次执行的 HPLOT 语句画线或点的终止坐标开始画一条直线到坐标（X_2，Y_2）；

④ HPLOT X_1，Y_1 TO X_2，Y_2 TO … TO X_n，Y_n

在（X_1，Y_1），（X_2，Y_2），…，（X_n，Y_n）之间画直线；

⑤ HPLOT TO X_2，Y_2 TO …TO X_n，Y_n

从上次执行的 HPLOT 语句画线或点的终止坐标开始画线到(X_2，Y_2)，再到其它坐标，直到(X_n，Y_n)。

注意之点：

a）要求 $0 \leqslant X \leqslant 279$，$0 \leqslant Y \leqslant 191$。

b）画线中若坐标间隔符合 $|Y_2 - Y_1| = |X_2 - X_1|$，则画出的是直线，否则带有小的波折线。

显示高分辨图形的例子

例 1：

```
10   HGR
20   HCOLOR= 7
30   HPLOT 20,40
40   HPLOT 0,0 TO 279,191
50   HPLOT  TO 0,80
```

运行这个程序，建立了高分辨图形第一页，画出一个小点，一条长对角线，最后画线到坐标(0,80)处，点、线颜色为白

色，屏幕底端因是正文显示区，画不出线条来，两条直线在右下端是断开的。

例 2：

```
10   HGR
15   POKE  - 16302,0
20   HCOLOR= 7
30   HPLOT 20,40
40   HPLOT 0,0 TO 279,191
50   HPLOT  TO 0,80
```

运行这个程序，显示结果与上例稍有不同，由于是全屏幕显示，两条直线完全显示出来。

2.13.3 TEXT 命令

执行 TEXT 命令将立即退出图形显示方式而进入全屏幕文本显示方式，每行40个字符，共24行，提示符移到屏幕最后一行，且不清除屏幕上原有的信息。

附录(一) ASCII 码

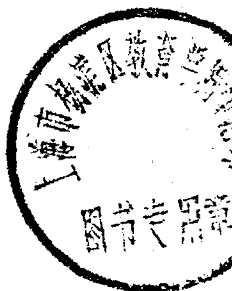
(美国信息交换标准代码)

ASCII 码	显示字	键
0		CTRL-@
1		CTRL-A
2		CTRL-B
3		CTRL-C
4		CTRL-D
5		CTRL-E
6		CTRL-F
7	(BELL)	CTRL-G

8	(BACK SPACE)	CTRL-H or ←
9		CTRL-I
10	(LINE FEED)	CTRL-J
11		CTRL-K
12		CTRL-L
13	(CARRIAGE RETURN)	CTRL-M
14		CTRL-N
15		CTRL-O
16		CTRL-P
17		CTRL-Q
18		CTRL-R
19		CTRL-S
20		CTRL-T
21	(FORWARD SPACE)	CTRL-U or →
22		CTRL-V
23		CTRL-W
24	(CANCEL LINE)	CTRL-X
25		CTRL-Y
26		CTRL-Z
27		ESC
28		na
29		CTRL-SHIFT-M
30		CTRL-
31		n.a.
32	SPACE	SPACE BAR
33	!	!
34	"	"
35	#	#
36	$	$
37	%	%
38	&	&
39	'	'
40	(	(
41	)	)
42	*	*
43	+	+
44	,	,

45	-	-
46	.	.
47	/	/
48	0	0
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	:
59	;	;
60	<	<
61	=	=
62	>	>
63	?	?
64	@	@
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
70	F	F
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O	O
80	P	P
81	Q	Q

82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W
88	X	X
89	Y	Y
90	Z	Z
91	[	[
92	¥	¥
93	]	]
94	^	^



注：如果可以由监督系统或整数BASIC产生，则加入128到ASCII码内。

附录（二） APPLESOFT

保留字及代表数字

代表数字	保 留 字	代表数字	保 留 字	代表数字	保 留 字
128	END	164	LOMEM:	200	+
129	FOR	165	ONERR	201	−
130	NEXT	166	RESUME	202	*
131	DATA	167	RECALL	203	/
132	INPUT	168	STORE	204	^
133	DEL	169	SPEED=	205	AND
134	DIM	170	LET	206	OR
135	READ	171	GOTO	207	>
136	GR	172	RUN	208	=
137	TEXT	173	IF	209	<
138	PR#	174	RESTORE	210	SGN
139	IN#	175	&	211	INT
140	CALL	176	GOSUB	212	ABS
141	PLOT	177	RETURN	213	USR
142	HLIN	178	REM	214	FRE
143	VLIN	179	STOP	215	SCRN(
144	HGR2	180	ON	216	PDL

续表

代表数字	保　留　字	代表数字	保　留　字	代表数字	保　留　字
145	HGR	181	WAIT	217	POS
146	HCOLOR=	182	LOAD	218	SQR
147	HPLOT	183	SAVE	219	RND
148	DRAW	184	DEF	220	LOG
149	XDRAW	185	POKE	221	EXP
150	HTAB	186	PRINT	222	COS
151	HOME	187	CONT	223	SIN
152	ROT=	188	LIST	224	TAN
153	SCALE=	189	CLEAR	225	ATN
154	SHLOAD	190	GET	226	PEEK
155	TRACE	191	NEW	227	LEN
156	NOTRACE	192	TAB(	228	STR$
157	NORMAL	193	TO	229	VAL
158	INVERSE	194	FN	230	ASC
159	FLASH	195	SPC(	231	CHR$
160	COLOR=	196	THEN	232	LEFT$
161	POP	197	AT	233	RIGHT$
162	VTAB	198	NOT	234	MID$
163	HIMEM:	199	STEP		

注：在下面情况下，APPLESOFT辨认不出TO为保留字：
① 在TO之前刚好是字符 A;
② T与O之间有空白。

附录(三)　DOS保留字

DOS 命令在直接使用时或在程序中，跟在打印语句的控制字后面时，也是保留字，有如下这些：

APPEND CHAIN INIT POSITION SAVE BLOAD CLOSE LOAD READ UNLOCK

BRUN DELETE LOCK RENAME
UERIFY BSAVE EXEC OPEN
RUN WRITE

第三章 MICROSOFT BASIC 语言

§ 3.1 CP/M操作系统

3.1.1 Z-80插件和CP/M系统盘

APPLE公司为了使APPLE Ⅱ机有更强大的功能，也能使用广泛流行的以Z-80为CPU的微机所适用的软件，设计了一种Z-80插件，其中含有一片Z-80A微处理机，把这种插件插入APPLE Ⅱ的扩展插槽，就使APPLE Ⅱ机变成了一个双CPU的微机，既能由DOS系统控制，也能由CP/M系统控制，这将由软件命令来选择6502 CPU操作方式或Z-80CPU操作方式。与Z-80插件一起提供的是一块CP/M系统盘，即由Digital Research公司所研制的CP/M-80磁盘操作系统，以及在这个操作系统支持下的MICROSOFT BASIC语言。

3.1.2 CP/M-80的版本

1973年Gary Kildall首先研制出CP/M操作系统，即控制/监督系统(Control Program/Monitor)，1981年得到了很大的发展，成为微型计算机中应用非常广泛的操作系统，为了适应各种情况的需要，发展出了多种版本。

CP/M版本表示法：在小数点左面的数字代表整体的型号区别，小数点右边的数字代表同一型号的修订版，小数点

右边的第二个数字代表版本上的细微差别或只与具体机型有关的版本。CP/M-80[1]的版本主要有：

1.3 最原始的CP/M-80版本；

1.4 版本1经修订后的版本；

2.0 版本2的原始型；

2.1 版本2经修改后的版本；

2.2 版本2的最近修订版。

这是按Digital Research公司对于CP/M的修改版号分类，还有的厂商又自行修改版本，则在右边第二位小数上表示出来。

这些操作系统版本影响微机功能主要有如下一些差别，分别见表1—3。

表 3.1 行编辑命令

命令	1.3 版	1.4 版	2.2 版
CTRL-C	有	有	有
CTRL-E	无	有	有
CTRL-H 或退格键	无	无	有
CTRL-J 或行进入键	无	无	有
CTRL-M 或回车键	有	有	有
CTRL-P	有	有	有
CTRL-R	无	有	改进
CTRL-S	有	有	有
CTRL-U	有	功能相同	保留命令行
CTRL-X	无		退格并删除命令行
DELETE 或CTRL-@	有	有	有

1) CP/M-80能使用于8080，8085和Z80的CPU；还有一种CP/M-86版本适用于8086和8088型的CPU。

表 3.2 新的或修改的命令

命 令	1.3 版	1.4 版	2.2 版
DIR	有	有	新的
	每行显示1个文件名称和类别	最大每行显示4个文件名称和类别	
ED	有	新的	新的
ERA	有	新的	新的
	不显示校验信号	显示检验信号（Y/N)?	
PIP	有	新的	新的
SAVE	有	有	新的
STAT	有	新的	新的
SUBMIT	有	有	新的
USER	无	无	有

表 3.3 使用磁盘的不同

项目	1.3 版	1.4 版	2.2 版
最多可用驱动器数目	2个	4个	16个
每个驱动器最大容量	1兆字节	1兆字节	16兆字节
最大可存文件数	64个	64个	可扩展
存取方法	顺序存取	顺序存取	顺序存取或随机存取
磁盘特性存贮单元	BDOS	磁盘参数块	BIOS

由于微机系统配置的不同，在使用CP/M-80操作系统时，还有多种不同格式的磁盘，因此用到具体的微机时，必须选定相应的磁盘：

44K系统磁盘：指的是用于RAM为48K的APPLE Ⅱ或APPLE Ⅱ PLUS上的磁盘；

56K系统磁盘：指的是带有语言卡插件；因而具有RAM为64 K的APPLE Ⅱ或APPLE Ⅱ PLUS系统上用的磁

盘；

13扇区磁盘：指的是使用APPLE DOS 3.2或更早版本，且无语言卡插件的微机上用的磁盘；

16扇区磁盘：指的是使用APPLE DOS3.3或有语言卡插件的微机上用的磁盘，除了包括13扇区磁盘中所有软件功能外，并且具备高分辨绘图功能的GBASIC语言。

我们在这一章介绍的是使用了语言卡插件的APPLE Ⅱ PLUS，并用的是APPLE CP/M2.2版操作系统，因而使用16扇区56K系统磁盘。

3.1.3 CP/M-80系统对微机外设的要求

CP/M-80操作系统要求连接外部设备的插件必须插在固定的插槽位置，这与APPLE DOS是不同的，而与APPLE PASCAL对外部设备的要求类似。

CP/M-80系统用字母及其后的冒号代表各台磁盘驱动器，下面所列举的CP/M-80所要求外部设备连接插槽位置时，对驱动器就是这样命名的。

下表是APPLE CP/M-80系统下外部设备必须连接的插槽位置：

插槽编号	外部设备
0	语言卡插件
1	打印机、视频终端插件
2	调制解调器、光电输入或凿孔机插件
3	控制台输出插件或视频终端
4	插磁盘驱动器插件、连接E：和F：磁盘机的盘控、或插Z-80插件
5	插磁盘驱动器插件连接C：和D：磁盘机控制器
6	插磁盘驱动器插件连接A：和B：磁盘机的控

制器（一定要插入）

7　　备用，适用于所有输出输入插件或插Z-80插件

正如上表指出的那样，在6号插槽至少应装一台磁盘驱动器，若还有别的驱动器插件，则依次应插在5号插槽，最后是4号插槽。

把插件插入插槽的步骤：

（1）必须关掉电源。

（2）双手拿住插件顶端垂直插入插槽，注意绝不要插错方向；不要用手玷污插件的插脚，以免接触不良；插件插入后，应前后轻轻晃动一下，以确保插件与插槽接触可靠。

（3）检查一遍无误，再开机。

3.1.4　建立56K的CP/M-80系统

如果APPLEⅡ上已经配置了语言卡插件，则必须把随微机带来的44K CP/M系统磁盘修改为56K的系统后，才能用上扩充了的内存，方法是：

（1）为了保存好随微机带来的系统盘，这项工作最好在复制的系统盘上做，复制方法见后面的介绍。

（2）把复制了的44K的CP/M系统盘插入A:驱动器，开机，显示A>。

（3）键入CPM56A：按回车键，微机即进入转换操作，最终建成了56K的CP/M系统盘。

用建立的56K的CP/M系统盘引导主机，这时就可使用语言插件16K RAM中的12K内存。CP/M占用7K RAM；CP/M和MBASIC一起占29K RAM；CP/M和GBASIC（高分辨率的绘图BASIC）一起占用37K多RAM。

3.1.5　CP/M-80系统盘介绍

APPLE Ⅱ PLUS微机有自动启动操作系统的功能，只要把CP/M-80盘插入A：驱动器，开机，即进入CP/M操作系统，提示符是A>。只要键入DIR，按RETURN键，就可见到CP/M系统盘上的组合程序目录，如下：

```
A>
A: FORMAT    COM : COPY     COM
A: MBASIC    COM : GBASIC   COM
A: CONFIGIO  BAS : PIP      COM
A: STAT      COM : ED       COM
A: ASM       COM : DDT      COM
A: LOAD      COM : RW13     COM
A: APDOS     COM : SUBMIT   COM
A: XSUB      COM : DUMP     ASM
A: DUMP      COM : DOWNLOAD COM
A: CPM56     COM
```

以下对该目录先作简单介绍，对其中其些重要的程序以后再作详细介绍。

(1) FORMAT 是对磁盘执行格式化（初始化）的应用程序。

(2) MBASIC 即MICROSOFT BASIC 解释程序。

(3) CONFIGIO 用来设定APPLE CP/M的操作环境，使与系统配置的外设相配合。

(4) STAT 提供磁盘状态信息的应用程序。

(5) ASM 为CP/M8080汇编程序，用于编写8080汇编语言程序。

(6) LOAD 用来将扩展字为.HEX的磁盘文件转换为机器可执行的 .COM文件；LOAD能将汇编程序的输出转换成机器代码。

（7）APDOS　可将程序文件和二进制文件从APPLE DOS磁盘转移到CP/M磁盘上去。

（8）XSUB　它与SUBMIT合用，在执行程序时，可随时从磁盘文件中输入字符。

（9）DUMP　可将磁盘文件内容以16进制数字型式显示出来。

（10）CPM56　把44K的CP/M系统磁盘转变为56K的系统磁盘的应用程序。

（11）COPY　用以复制CP/M磁盘，或将规格化的磁盘制成空白的CP/M系统磁盘。

（12）GBASIC　具有高分辨绘图功能的BASIC解释程序，其它功能与MBASIC相同。

（13）PIP　用于磁盘之间文件的转移，或将磁盘文件转移到终端或打印机。

（14）ED　是CP/M系统的文本编辑程序，可以对文本缓冲区的内容进行插入、删除。

（15）DDT　为CP/M的动态调试的程序，用以检测、修改8080汇编语言的程序错误。

（16）RW13　能用16扇区CP/M存取13扇区CP/M磁盘上的文件，若与PIP合用，并有两台或以上磁盘机，可将13扇区磁盘上的文件转移到16扇区的磁盘上。

（17）SUBMIT　能通过磁盘文件中的命令和程序执行操作，达到自动处理目的。

（18）DUMP.ASM　是DUMP程序的源程序列表，用汇编语言写成。

（19）DOWNLOAD　用于两台都是CP/M的微机之间通讯（通过R-232串行数据连接接口）。

§ 3.2 磁盘格式化和复制

3.2.1 进入CP/M系统步骤

由于APPLE Ⅱ PLUS微型机具有自动引导的ROM，所以引导CP/M系统很简单，如下：

(1) 把CP/M系统盘插入A：驱动器。

(2) 开机，驱动器红灯亮，CP/M系统自动进入主机，屏幕显示：

```
APPLE  Ⅱ  CP/M
56K  VER.  2.20B
(C) 1980  MICROSOFT
A>
```

其中A>就是CP/M系统的提示符。若是键入B：回车，则转到B：驱动器。若是要求控制打印机，就必须调用系统盘的DDT程序，先键入DDT，按回车键，则屏幕显示（以下为了表达方便起见，屏幕显示用大写表示，键盘输入用〈〉括起来，按回车键用〈CR〉表示）：

```
A> DDT    〈CR〉
DDT  VERS  2.2
-〈SDD2F〉
DD2F  3E  〈31〉
DD30    DD
```

这时再按CTRL-C(即同时按CTRL键和C键)，则启动了A：驱动器（称热引导)，回到提示符A>，再按CTRL-P，回车，即已连通打印机。这时，若是要打印系统盘上目录，可以键入打印目录命令DIR，按回车键，就能在打印机上输出主盘上所有程序的目录。再按CTRL-P键，脱离打

印机输出。

若是所用的APPLEⅡ机不具备自动导入功能，则开机后，先按RESET键，再按6CTRL-K，然后按回车键，就进入CP/M系统了。

3.2.2 磁盘格式化

新的空白磁盘在使用之前也必须先做格式化的工作，有如下两种方法：

(1) 用一个驱动器做格式化工作

假如只有一个驱动器，则步骤是：

① 把系统主盘插入A:，开机；

② 键入FORMAT A:（需要注意的是，CP/M 系统与DOS不同，命令与执行对象之间必须要有空格，若是不留空格，则无法识别命令），则显示：

APPLEⅡ CP/M

16 SECTOR DISK FORMATTER

(C) 1980 MICROSOFT

INSERT DISK TO BE FORMATTED IN DRIVE A:

PRESS RETURN TO BEGIN

即要求把空白磁盘插入A:，然后按回车键执行初始化。按回车键后，若确实是空白盘，则显示FORMATTING…，磁盘驱动器红灯亮，过一会儿（由于磁盘旋转稳定要求，此段时间约需30秒），红灯灭，显示：

FORMAT COMPLETE

INSERT CP/M SYSTEM DISK IN DRIVE A:

PRESS RETURN

表示格式化工作已经完成，要求插入系统主盘，按回车键，以回到A＞。

假如要求格式化的盘并不是新的空白盘，而是已存有资料的旧盘，执行格式化将先把旧盘上的资料删去，这时当发出了FORMAT命令之后，显示屏显示：

DISK IN DRIVE A:WILL BE ERASED

CONTINUE (Y/N)?

即主机提醒使用者：再往下执行就要删除磁盘上的信息了，请再检查一下，若确属需要删除内容的磁盘，要继续下去，则按Y键。屏幕显示FORMATTING…，驱动器红灯亮，过一会儿，红灯灭，显示：

FORMAT COMPLETE

INSERT CP/M SYSTEM DISK IN DRIVE A:

PRESS RETURN

即说明格式化已经完成，要求插入系统主盘，按回车键，以回到A＞。

(2) 用两个驱动器做格式化工作

若是有两个驱动器，做格式化工作步骤：

① 系统主盘插入A:，开机，进入A＞；

② 键入FORMAT，回车，则显示：

APPLE Ⅱ CP/M

16 SECTOR DISK FORMATTER

(C) 1980 MICROSOFT

FORMAT DISK IN WHICH DRIVE?

表示已进入格式化程序，问要求格式化的空盘放在哪个驱动器？

③ 若键入B：回车则显示：

INSERT DISK TO BE FORMATTED IN DRIVE B:

PRESS RETURN TO BEGIN

要求把空白盘插入驱动器B:，并按回车键，这时又会有两种情况：若确定是新的空白盘，则显示：

FORMATTING…

驱动器红灯亮，不久即完成格式化，显示询问：

FORMAT COMPLETE

FORMAT DISK IN WHICH DRIVE?

若是存有数据的旧盘，则是另一种情况，按了回车键，显示的是：

DISK IN DRIVE B:WILL BE ERASED

CONTINUE (Y/N)?

要求重新考虑是否做格式化工作，若继续做，则按Y键，显示：

FORMATTING…

驱动器红灯亮，不久即完成格式化，这时才显示：

FORMAT COMPLETE

FORMAT DISK IN WHICH DRIVE?

问下一次做格式化的盘放在哪个驱动器，以再做另一磁盘的格式化工作，若是不做了，则按回车键，又回到A〉。

CP/M-80系统下做了格式化工作的磁盘已经可以存贮资料，但是在格式化磁盘上并未置入CP/M系统，因此不能用来引导主机，这一点是与APPLE DOS系统不同的。对于CP/M-80系统下执行了格式化的磁盘必须再用COPY程序把CP/M系统由系统盘装入，才能执行引导工作。

3.2.3 复制磁盘

(1) 用一个驱动器做复制工作步骤：

① 系统主盘插入A:，开机，进入A＞。

② 键入COPY A:＝A:　回车则显示：

APPLE Ⅱ CP/M

16[1)] SECTOR DISK COPY PROGRAM

(C) 1980 MICROSOFT

INSERT MASTER DISK AND PRESS RETURN

要求插入“源”盘（指要求复制的原始盘），按回车键。则又显示：

INSERT SLAVE DISK AND PRESS RETURN

要求插入空白盘（已格式化的），并按回车键，则又显示：

INSERT MASTER DISK AND PRESS RETURN

又要求插入“源”盘，按回车键。等等，如此重复多遍，最后完成显示：

COPY COMPLETE

DO YOU WISH TO MAKE ANOTHER COPY?

表示复制完成，若还复制，按Y键；否则按N键，显示：

INSERT CP/M SYSTEM DISK INTO DRIVE A:

HIT RETURN

要求插入CP/M盘，回车，以回到A＞

---

1) 若用13扇区磁盘，则此处为13。

用一个驱动器做复制工作需要反复插换磁盘，较麻烦，有两个驱动器做复制工作就较方便了。

(2) 用二个驱动器做复制工作步骤：

① 系统主盘插入A:，开机，进入A＞

② 键入COPY B:=A: 回车则显示：

APPLE Ⅱ CP/M

16 SECTOR DISK COPY PROGRAM

(C) 1980 MICROSOFT

INSERT MASTER DISK INTO DRIVE A:

INSERT SLAVE DISK INTO DRIVE B:

PRESS RETURN TO BEGIN

要求把“源”盘插入A:，空白盘插入B：并按回车键，这时将A:中内容写入B:，显示：

COPYING…

最后显示：

COPY COMPLETE

DO YOU WISH TO MAKE ANOTHER COPY?

表示复制完成，若要求继续复制另一个磁盘，则键入Y，又重复复制过程；否则键入N，则显示：

INSERT CP/M SYSTEM DISK INTO DRIVE A:

HIT RETURN

要求把CP/M系统盘插入A:，按回车键，以回到A＞。可以看出，双驱动器的同步复制过程要简单得多。

(3) 灵活做法

也可以先不指定用几个驱动器做复制工作，而在复制操

作过程中指定，步骤是：

① 把CP/M系统主盘插入A:，开机，进入A〉。

② 键入COPY，回车则显示：

APPLE Ⅱ CP/M

16 SECTOR DISK COPY PROGRAM

(C) 1980 MICROSOFT

*

根据在提示符*后面键入驱动器名字，可用一个或几个驱动器做复制工作，其复制过程及操作步骤与上面介绍的复制方法类似，不多介绍了。

3.2.4 复制CP/M系统

上面介绍格式化时已指出，格式化后的磁盘并不包含CP/M操作系统，因而不能用以引导主机。以下介绍用两个磁盘驱动器把CP/M操作系统复制到这些磁盘上的办法，这时对磁盘上的其它文件没有影响。复制CP/M系统的步骤是：

(1) 系统主盘插入A:，开机进入A〉。

(2) 键入COPY B:=A:/S 回车则显示：

APPLE Ⅱ CP/M

16 SECTOR DISK COPY PROGRAM

(C) 1980 MICROSOFT

INSERT MASTER DISK INTO DRIVE A:

INSERT SLAVE DISK INTO DRIVE B:

PRESS RETURN TO BEGIN

要求在A:插入CP/M盘，在B:插入被复制盘，并按回车键。照此办理，则显示：

COPYING…

最后显示：

COPY COMPLETE

DO YOU WISH TO MAKE ANOTHER COPY?

表示复制完成，若要复制另一个磁盘，则按Y键；否则按N键，则显示：

INSERT CP/M SYSTEM DISK INTO DRIVE A:

HIT RETURN

要求把CP/M系统盘插入A:按回车键，回到A〉。

若是只用一个驱动器，也可复制，方法类似前面介绍过的复制“源”盘的方法。

3.2.5 错情信息

做格式化和复制操作时，可能会有如下的出错信息：

DISK WRITE PROTECTED

磁盘已采取保护措施，不能存信息；

DIST I/O ERROR 驱动器也许没有关好，磁盘没有放好或没有放磁盘；

COMMAND ERROR 命令写错。

§3.3 键盘用法

3.3.1 一般用法

在CP/M系统控制下键盘的用法与DOS有很多是不同的，用于程序编辑时更有很大的不同。现将用法介绍如下：

← 为后退键，并且删除光标所指该字符；

CTRL-E 屏幕上光标移到下一行的开头，再按回车键，下一行的语句才进入内存；

CTRL-H　作用与←键相同；

CTRL-J　中止输入，与回车键作用相同；

CTRL-R　重新显示屏幕正显示的这一行；

CTRL-X　光标退到这一行的开头，整行字符全删掉；

CTRL-K　左边的方括号；

CTRL-B　向后斜线(\)

3.3.2　控制输出的键

CTRL-S 暂时中止屏幕字符输出，再按下任何一个键，又恢复输出；

CTRL-P 若已用 DDT 程序使微机控制了打印机，则以后用到列目录等命令时，只要再按这两个键，再按回车键，就能在打印机上输出，中间若再按这两个键，又停止输出。

3.3.3　冷引导和热引导

每次开机把CP/M系统引导入主机的动作称为冷引导；在操作过程中的某一时刻，按键CTRL-C也会引导磁盘上的CP/M系统进入主机，称热引导，用这种方法随时保证内存中CP/M系统处于工作状态而不破坏内存中的资料。当每次在驱动器上更换磁盘时，都要求做热引导，所以这是要经常用到的操作。假如在驱动器中已换了磁盘，而没有执行热引导，若这时要把资料存入磁盘就会出错，显示：

BDOS ERR ON X：DISK R/O

(X：是驱动器的字符编号)。表示磁盘换入后未键入CTRL-C，或因防护标志使磁盘不能写入。这时只要按回车键，则也执行热引导，主机回到A＞。

有些程序在运行结束时，自动执行热引导，这时换磁盘就不必再执行热引导了。

在CP/M系统控制下，并有自动起动功能时，按下CTRL-RESET键，也执行热引导。

§ 3.4 CP/M的直接命令

3.4.1 CP/M的文件命名规则

在介绍CP/M直接命令之前，要先了解一下CP/M的文件命名规则。因为CP/M系统文件的命名规则与 APPLE DOS 有所不同。现将主要规则介绍如下：

(1) 每个CP/M文件名由两部分组成，即文件名和扩展名，中间用小数点符号隔开。文件名必须大写，也可以用数字，最多只能用8个字符。扩展名可以不用，最多只能用3个字符，扩展名一般表示这个文件内部格式。例如有如下一些CP/M文件名：

FILE.BAS　文件名FILE，扩展名BAS

86000.COM　文件名86000，扩展名COM

A-12.ASM　文件名A-12，扩展名ASM

现举出扩展名所代表的文件格式例子：

ASM　表示用汇编语言写成的文件

ASC　包含ASCII字符文件

BAS　表示BASIC语言文件

BAK　表示备用文件

COM　表示可以立即执行的过渡文件

DAT　表示存贮资料的文件

FOR　表示FORTRAN语言文件

(2) 以下一些字符不能用在文件名中：

< > · , ; : = ? * [ ] ( ) / 或 <TAB>

(3) 不可使用控制字符或其它无法在屏幕上显示的字符

作为文件名。

3.4.2 成组访问文件

CP/M提供了成组访问磁盘文件的功能，可以提高磁盘操作效率，主要用两种方法：

(1) 用符号*代表文件名或扩展名。

*.*代表磁盘上所有文件

*.COM代表磁盘上所有扩展名为COM的文件

FILE.*代表所有名字为FILE的文件

(2) 用符号？代表文件名或扩展名中个别字符。例如：

A?B.BAS 指磁盘上除？处字符不同以外的所有同名文件

ABC.??? 表示所有名字为ABC的文件

????.??? 与*.*作用相同。

3.4.3 CP/M的直接命令

CP/M的直接命令也称内部命令，它可在提示符A＞之下直接由CP/M执行的命令，而不必再从磁盘上取出，以下只介绍几种常用的命令。

DIR 命令

是显示磁盘文件目录的命令。

DIR 显示目前驱动器中文件目录；

DIR B:显示驱动器B:中文件目录；

DIR ABC.BAS 显示文件名为ABC，格式为BAS的文件目录；

DIR B:CDE.* 显示驱动器B:中文件名为CDE的所有文件目录。

前面已介绍过，在引入CP/M后若执行了DDT连接打印机命令，则在键入DIR，并按下CTRL-P，再按回车键，

就可以在打印机上输出指定文件目录。

ERA 命令

为删除磁盘文件的命令，也有多种格式：

ERA B: ABC. BAS　删除驱动器 B: 中指定文件名；

ERA DEFG.*删除现用驱动器中文件名为 DEFG的所有文件

ERA *.*删除现用驱动器中所有文件，此时按下回车键后显示提问：

ALL(Y/N)?

提示是否要删除所有的文件，假如坚持删去，则按Y 键及回车键。

REN 命令

为重新命名磁盘文件的命令，格式是：

REN　新文件名=老文件名

文件名之前也可指定驱动器，例如：

REN　B: ABC. BAS=DDD. BAS 把驱动器 B:中的DDD.BAS 文件名改为ABC. BAS，即新文件名列在等号的左边，旧文件名列在等号的右边。

TYPE 命令

是显示以 ASCII 码贮入的资料文件内容的命令，例如：

TYPE　B: EFG. BAS　指定驱动器B:，把文件 EFG. BAS的内容列出。

直接命令的错误信息

① NO FTLE, NOT FOUND或FILE NOT FOUND 表示磁盘上没有命令所要求的文件。

② BDOS ERR ON X:
(X: 代表驱动器名字，如B:)。表示CP/M 无法找到所指定的磁盘或驱动器，说明磁盘装置有问题，也许是没放磁盘，磁盘没有放好或磁盘没做过格式化等。

BDOS ERR ON X: BAD SECTOR
表示磁盘装置有问题。键入CTRL-C 则执行热引导。

BDOS ERR ON X: DISK R/O
表示磁盘装置有错，原因是:

a) 换了磁盘后没有执行热引导;

b) 磁盘是写保护的。

这时按动任何一个键，则会热引导，回到A〉。

BDOS ERR ON X: FILE R/O
表示指定的磁盘文件原先用STAT 程序指定为只读文件，不能执行写入命令。这时若键入任一字符，则执行热引导，回到A〉。

BDOS ERR ON X: SELECT
不存在指定的驱动器。这时若键入任一字符，则执行热引导，回到A〉。

③ FILE EXISTS 在执行REN命令时，磁盘中已存在与指定新文件名同名的文件。

§ 3.5 CP/M的过渡命令

CP/M的过渡命令(transient command)与直接命令不同，其内容是贮存在磁盘上的程序或命令(文件格式是COM)，执行前先得从磁盘上引入，再执行。只是键盘输入命令的方法与输入直接命令方法一样，即在提示符A〉之后键入命令即可。命令行最多容许127个字节。以下介绍主要的

过渡命令：

3.5.1 FORMAT 命令

为格式化命令，前面已介绍过。

3.5.2 COPY 命令

为复制整片磁盘文件命令，前面已介绍过。

3.5.3 CPM 56 命令

转变44 K的CP/M系统盘为56 K的CP/M系统盘，前面已介绍过。

3.5.4 STAT 命令

STAT(Statistics)命令主要有两类使用功能：

(1) 统计显示一个或一组磁盘文件的情况，用法如下：

① STAT 〈CR〉 (〈CR〉代表回车键)

这个命令能够把主机起动以来所用驱动器中磁盘的情况显示出来。例如：

A 〉STAT 〈CR〉

A: R/W, SPACE: 2 K

表示驱动器A：中磁盘为读/写磁盘，还有2 K的存贮空间可供使用（存贮空间的单位为字节，下同）。

又例如：

A 〉STAT 〈CR〉

A: R/W, SPACE: 3 K

B: R/O, SPACE: 40 K

表示驱动器A：中磁盘为读/写磁盘，还有3 K的存贮空间；B：中为只读磁盘，还有40 K存贮空间。

② STAT X: 〈CR〉

显示驱动器X：中磁盘有用存贮空间。例如：

A 〉STAT B: 〈CR〉

BYTES REMAINING ON B: nnnK

表示在B：驱动器中磁盘还有nnnK字节的存贮空间。

③ STAT X：文件名〈CR〉

显示驱动器X：中磁盘上指定文件的大小和属性。例如：

A〉STAT ABCDE. ASC〈CR〉

显示：

RECS	BYTES	EXT	ACC	
5	3K	1	R/W	A：ABCDE. ASC

BYTES REMAINING ON A：100K

表明文件ABCDE. ASC占有5个记录（RECS)，长度3K字节（BYTES)，占1个实际段落（EXT，每一个实际段落占有16K字节的磁盘空间)，文件存取属性(ACC)为读写型。

又例如：

A〉STAT B：*. COM〈CR〉

显示：

RECS	BYTES	EXT	ACC	
4	1K	1	R/W	A：DUMP. COM
52	7K	1	R/W	A：(ED. COM)
58	8K	1	R/W	A：PIP. COM
48	6K	1	R/W	A：STAT. COM
10	2K	1	R/W	A：SUBMIT. COM

BYTES REMAINTING ON A：200K

表中列出各文件所占记录个数、长度，所占实际段落数，指明文件是只读（R/O）还是读写（R/W)，最后说明磁盘上还有200K的存贮空间可供使用。其中A：(ED. COM)表示这个文件不会出现在目录中，是SYS型文件(后面介绍)。

④ STAT X：文件名＄指定属性〈CR〉

CP/M-80的2.2版指定文件两对属性：

R/O与R/W：R/O是只读型文件，只能读入这个文件，不能改变或删除它；R/W是读写文件，可读、可改、可删。平常的文件都是R/W型的，除非特别规定。

SYS与DIR：SYS为系统型文件，这个文件不出现在目录中（即执行DIR命令后不显示），但可以修改、删除或读出这个文件，只有使用STAT *.*命令才会看到它的目录；DIR为可列目录文件，可以显示出来，一般文件属于这一类。

文件属性可用命令指定。例如：

A〉STAT QWZ. BAS ＄R/O〈CR〉

显示QWZ. BAS SET TO R/O

表示指定名为QWZ. BAS的文件为只读文件。冷引导或热引导不改变它的只读性质。

又例如：

A〉STAT *. * ＄R/W〈CR〉

显示：QWZ. BAS SET TO R/W

ABCD. COM SET TO R/W

表示指定文件都为读/写型文件。

又例如：

A〉STAT B：EXAMPLE. BAS ＄ SYS 〈CR〉

显示：EXAMPLE. BAS SET TO SYS

表示指定文件EXAMPLE. BAS为系统文件。这时再用DIR列目录，已看不到这个文件名字。

（2）统计外部设备配置信息

这个命令可以提供有关CP/M所用外部设备情况，其主要用法如下：

① STAT X:=R/O 〈CR〉

表示指定X：驱动器中磁盘为暂时的只读状态，这时所有的磁盘文件都不能修改或删除。假如执行冷引导或热引导，就可解除只读状态。

② STAT DSK: 〈CR〉

要求显示当前驱动器X：中磁盘特性，这时会显示（X为任选的驱动器，下例中X指定为A驱动器）：

```
A>STAT DSK:

   A: Drive Characteristics
1024: 128 Byte Record Capacity
 128: Kilobyte Drive  Capacity
  48: 32  Byte Directory Entries
  48: Checked  Directory Entries
 128: Records/ Extent
   8: Records/ Block
  32: Sectors/ Track
   3: Reserved Tracks
```

表中依次列出了：磁盘共有的记录总数；磁盘总容量（K BYTE)；最多可存贮48个文件；每个目录入口至多可以有128个记录；文件定位时，至少要求8个记录的磁盘空间；每个磁道分成32个扇区；有3个磁道不能存文件。本例中48: Checked Directory Entries一行，由于是用可更换的磁盘，因此与上一行意义相同。若系统配置硬盘，此行为0。

这些磁盘特性与所用的CP/M系统有关。

③ STAT DEV: 〈CR〉

显示：

```
A>STAT DEV:
CON: is CRT:
RDR: is PTR:
PUN: is PTP:
LST: is LPT:
```

这个命令把微机外部设备配置情况显示出来，左边代表逻辑装置，右边代表指定给逻辑装置的实际外部设备。在CP/M里，有4种逻辑装置，代表微机4种功能，即：

CON：接收命令，显示信息，为主机控制台功能；

RDR：接收信息，读入纸带功能；

PUN：输出信息，纸带凿孔功能；

LST：输出、列表功能。

有12种外部设备可分别配合实现这4种功能，即：

TTY：慢速主控机(电传打字机)；

CRT：快速主控机(CRT显示器)；

BAT：批量处理器；

UCT：用户定义的主控机；

PTR：纸带读入机；

PTP：纸带凿孔机；

UR1：用户读入机#1；

UR2：用户读入机#2；

UP1：用户凿孔机#1；

UP2：用户凿孔机#2；

LPT：行打印机；

UL1：用户列表装置。

上述4种逻辑装置与实际装置可作相应的配合，例如上面这个例子的情况表示：

接受命令和显示信息用显示器；

接受信息用纸带读入机；

输出资料用纸带凿孔机；

输出、列表用行式打印机。

这种配置方式对于指定的CP/M系统，出厂时已经指定，当然亦可用STAT命令改变对应的配置情况。

④ STAT log: = phy: 〈CR〉

指定实际外部设备（phy:）至逻辑装置（log:），以改变微机外部设备配置情况。在一个命令行里可以同时指定几个外部设备的改变情况，中间用逗号隔开，格式是：

STAT log: = phy:, log: = phy: 〈CR〉

⑤ STAT USR: 〈CR〉

可显示用户号及正在操作的磁盘文件号，例如：

```
A>STAT USR:

Active User : 0
Active Files: 0
```

表明当前用户号码为零，目前操作磁盘上，文件用户号码为零。

⑥ STAT VAL: 〈CR〉

可显示所有设备分配情况及STAT命令使用的情况，例如：

```
A>STAT VAL:

Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status  : DSK: d:DSK:
User Status  : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

表明 STAT命令用法的简单提示。例如指明键入 STAT d:= R/O可以使 d:驱动器中磁盘为暂时只读型磁盘；而键入 STAT d: DSK:〈CR〉，可以设定 d:驱动器的特性等等。

3.5.5 PIP 命令

PIP 是 Peripheral Interchange Program（外围交换程序）的缩写，主要用于文件的复制。有两种使用 PIP命令的方法：

PIP〈CR〉

显示 PIP 的系统符号＊，然后键入命令行：

＊PIP 的命令行〈CR〉

这时就会执行 PIP命令，执行完了又回到提示符＊，还可键入 PIP 的有关命令。所以这种方式适用于多次执行 PIP 命令的作业。若按了CTRL-C或回车键，则回到 A〉。

PIP 命令行〈CR〉

这是直接执行 PIP命令的方式，适用于一次性作业，作业错误则回到 A〉。

上述两种方式 PIP 命令的具体用途是：

(1) 把文件复制到磁盘上

① PIP X：新文件名 = Y：老文件名[P]〈CR〉即把驱动器 Y：磁盘上的老文件复制到驱动器 X：的磁盘上，换成新名字，使用参数 P[1]，参数 P 与文件名之间不能有空格。例如：

A〉PIP ABCD. BAS = DEF. BAS〈CR〉

这是在同一磁盘上把DEF. BAS文件复制为名字是 ABCD.

---

1) PIP 参数种类较多，待读者熟悉 BASIC语言可再行了解，此处从简。

BAS 的文件；

A〉PIP B：ABCD．BAS＝A：DEF．BAS 〈CR〉

把 A：中 DEF．BAS 文件复制到 B：中磁盘上；

A〉PIP B：ABCD．BAS＝A：DEF．BAS［V］〈CR〉

把 A：中 DEF．BAS 文件复制到 B：中磁盘上，名字改为 ABCD．BAS，参数 V 表示复制结束后，要求自动核对新老文件是否一致，若完成无误，则出现 CP/M 系统提示符。复制文件时，用这种办法以检查复制是否正确，应经常采用；

② 成批复制文件

例如：

A〉PIP B：＝A：*．BAS［V］〈CR〉

显示：

COPYING—

ABCD．BAS

CEF．BAS

HEJ．BAS

A〉

即是把 A：中扩展名为 BAS 的文件都复制到 B：中，文件名不变。复制过程中，自动显示复制的文件名；

③ 几个文件连接后复制，并建立新文件。例如：

A〉PIP B：ABC．BAS＝A：DE．BAS，B：FGH．BAS［V］〈CR〉

即把 A：中的 DE．BAS 文件与 B：中 FGH．BAS 连接起来，复制到 B：上成为新文件 ABC．BAS。

(2) 沟通磁盘文件与其他外部设备关系，或沟通外部设备间关系。例如：

A ＞ PIP LST：= A： ABC. BAS 〈CR〉

把A：中磁盘上文件 ABC. BAS 从打印机输出，按任何键则中止输出。

3.5.6 有关过渡命令的错误信息

一般性的错误信息

键入命令？可能是命令打错了。

BDOS ERR ON d：BAD SECTOR可能磁盘没有格式化，或磁盘上某一扇区有问题，或磁盘没有放好。

BDOS ERR ON d：SELECT驱动器指定错了，或驱动器门未关好，或该驱动器的电源未打开。

BDOS ERR ON d：R/O磁盘写保护，或换了磁盘以后没有执行热引导。

STAT 命令的错误信息

** ABORTED ** STAT命令终止了。

Bad Delimiter标点位置有误。

Invalid Assignment指定设备的格式有错。

Invalid File Indicator指定文件的格式有错。

** TOO MANY FILES ** 超过 STAT所能重排的文件数量的上限。

Invalid Disk Assignment磁盘指定方式不正确。

Wrong CP/M Versiion由于STAT的型号与CP/M型号不符而出错。

PIP命令的错误信息

DISK READ ERROR读磁盘错。

DISK WRITE ERROR写磁盘错。

VERIFY ERROR常与BDOS错同时发生。

NOT A CHARACTER SINK不能把字符送至该处。

READER STOPPED读入装置停止。

NOT A CHARACTER SOURCE不能从该处取得字符。

ABORTED处理过程终止了。

BAD PARAMETER[ ]内的参数不正确。

INVALID USER NUMBER用户编号不正确。

RECORD TOO LONG记录长度太大。

INVALID DIGIT与Hex格式不相符合。

END OF FILE, CTL-Z? PIP已经侦测到文件的末尾，但要求操作CTrl-Z键，以便加以肯定。

CHECKSUM ERROR表示Intel格式的检查值与读入的记录不符合。

CORRECT ERROR改正错误。

INVALID FORMAT格式不正确。

NO DIRECTORY SPACE文件目录已经满了。

NO FILE不存在指定的文件。

START NOT FOUND没有找到指定的起始字符串。

QUIT NOT FOUND没有找到指定的终止字符串。

CANNOT CLOSE FILE检查磁盘。

DESTINATION IS R/O目的文件为只读型，请按Y键删除它，并写入新的文件。

NOT DELETED对于上面的错误，若按N键，则显示此信息。

NOT FOUND与NO FILE类似。

REQUIRES CP/M2.0 PIP型号必须符合CP/M的型号。

UNRECOGNIZED DESTINATION 指定的外设不存在。

CANNOT WRITE无法写入。

INVALID PIP FORMAT检查标点是否有错。

CANNOT READ无法读出。

INVALID SEPARATOR检查标点是否有误。

§ 3.6 MICROSOFT BASIC语言

3.6.1 进入MBASIC的工作方式

MICROSOFT BASIC是为Z-80和8080微处理机设计的高级语言，目前已发行到第五版，使用极为广泛。MBASIC是MICROSOFT BASIC（也称BASIC-80）的CP/M版本，APPLE Ⅱ微机有了Z-80插件和CP/M系统盘就可用MBASIC语言编制程序。因为CP/M系统盘上的解释程序文件名称为MBASIC.COM，微机在CP/M系统控制下，出现A〉，键入过渡命令MBASIC，并按回车键，就能进入MBASIC工作方式；16扇区的CP/M系统主盘上除了MBASIC.COM程序文件外，还有GBASIC.COM程序，在CP/M系统提示符下键入GBASIC，并按回车键，就进入有高分辨图象显示的GBASIC工作方式。以下我们主要介绍第五版BASIC-80工作方式。进入MBASIC工作方式的格式有如下几种：

A 〉MBASIC〈CR〉

键入以上命令，就会显示：

```
BASIC-80  REV.5.2
[APPLE CP/M VERSION]
CORYRIGHT (C) 1980  BY MICROSOFT
CREATED: 12-NOV-80
26483 BYTES FREE
OK
```

(注：上述显示仅为例子，由于版本不同，建立时间不同等，具体日期、数字均会有所不同。)这时已进入MBASIC 工作方式，并且指定执行程序时最多只能同时打开3个文件，容许全部内存从CP/M中FDOS的起始位置开始，并指定随机文件记录的最大长度为128个字节。

A 〉MBASIC文件名〈CR〉

这种方式使用所有的内存，最多可同时打开3个文件，装入并立即执行指定的文件。

A 〉MBASIC文件名/F：文件数目〈CR〉

这种方式使用所有内存，最多可同时打开指定数目的文件，装入并立即执行指定的文件。

A 〉MBASIC文件名/F：文件数/M:内存的最大位置/S：记录的最大长度〈CR〉

这种方式使用户有较大选择，即指定了MBASIC使用内存的最大位置；允许同时打开的文件数；随机文件记录的最大长度以及装入并运行指定的文件。

3.6.2　程序语句中程序行、常数、字符和变量的格式

程序行

MBASIC中对程序行的用法与APPLESOFT类似，只是允许行号的范围是0—65529

常数

有五种类型：

① 整数：数值在－32768与32767之间；

② 单精度数：

表示范围：绝对值在 1×10^{-38} 至 1.701411×10^{38} 之间。单精度数指的是符合以下条件之一的数据：

a）长度≤7位的数（进入主机内存实际只有6位，故输出只有6位有效数字）；

b）指数形式是E；

c）数据尾部有！符号。

例如下面的程序及其运行输出结果：

```
10 A=123456!
20 LPRINT"A=";A
30 B=1.23457E+06
40 LPRINT"B=";B
50 C=111
60 LPRINT"C=";C
```

```
A= 123456
B= 1.23457E+06
C= 111
```

③ 双精度数

表示范围：基本与单精度数一致。

双精度数指的是符合以下条件之一的数据：

a）长度≥8位的数（向计算机输入数据时，双精度数的有效数位最多为17位，输出时只能有16位）；

b）指数形式是D；

c）数据尾部有#符号。

例如下面的程序及其运行结果：

```
10 A#=1.23456789123456 8D+17
20 LPRINT"A#=";A#
30 B#=12345678#
40 LPRINT"B#=";B#
50 C#=12.345#
60 LPRINT"C#=";C#
```

```
A#= 1.23456789123456 8D+17
B#= 12345678
C#= 12.345
```

④ 十六进制常数；数字前面以&H为标记；

⑤ 八进制常数，数字前面以&O或&为标记。

字符

与APPLESOFT的规定相同。

变量

MBASIC变量名的最大长度是40个字符，第一个字符必须是字母，变量名不能用保留字。字符串变量尾部以$为标记。数组变量最多可以有255维，每维最多可有32767个元素。

MBASIC的数值变量有三种形态，在使用之前指定，指定的方法有两种：

① 变量名最后为特定类型说明符：

% 整变量

! 单精度变量

双精度变量

若变量名后面没有以上符号，则默认为单精度变量。

② 用定义语句定义

a) DEFINT 字母域

在这个语句后的变量，只要是指定字母域中任一字母开头的变量就是整型变量，但若变量已有类型说明符，则以说明符规定的类型为准。例如：

10 DEFINT A，I，K—N

表示在10行以后，凡名字以A， I，K，L，M，N字母开头的变量都是整型变量，例如A1，L4都是整型变量。但是I#则是双精度变量，A4！则是单精度变量。

b) DEFSNG字母域

是定义单精度变量的语句，其它意义与以上对整型变量定义语句类似。

c) DEFDBL 字母域

是定义双精度变量的语句，其它意义与以上定义类似。

d) DEFSTR 字母域

是定义字符串型变量语句，其它意义与以上定义类似。

选择变量类型的原则：

选择变量的类型时，要考虑到运算精度、所占内存和运算速度三方面：

精度：以双精度数最高，整型数最差；

内存：双精度数要求8个字节内存，单精度数要求4个字节内存，整型数要求2个字节内存，所以双精度数占内存最多；

运算速度：双精度最慢，整型数最快。

另外，在程序中，同一个变量名，定义为不同的类型时，机器将认为它们是不同的变量，例如A#，A！，A%是不同的变量。

3.6.3 数的类型转换

在需要时，一个数值常数可从一种类型转换为另一种类型，可有以下几种情况：

（1）单精度数或双精度数转换为整型数时，按照数值取整原则进行。

（2）整型数转换为单精度数或双精度数时，所得数就是这个整型数所具有的精度。

（3）双精度数转换为单精度数时，最后一位有效数字是四舍五入得来的。

（4）单精度数转换成双精度数时，只有前面七位有效数是精确的，若单精度数的有效数字不到七位，可能有较大误差。例如：

```
10 ABCD#=1/3
20 ABCDEF#=234
30 ABCDEFGHIJ#=.369
40 LPRINT ABCD#
50 LPRINT ABCDEF#
60 LPRINT ABCDEFGHIJ#
```

运行后输出结果是：

```
.3333333432674408
234
.3690000176429749
```

（5）两个整型数算术运算结果若超出整型数范围，则结果作单精度数处理；除此以外，所有数的算术运算结果都以操作数中最高精度数处理，即运算结束后也回复到这个精度。

（6）数的关系运算中，计算机先把操作数都转化为运算数中较高类型的一类，然后计算。

（7）逻辑运算时，计算机先对操作数取整，然后运算。若取整超出范围，则为溢出错。逻辑运算的结果总是整数。

3.6.4 表达式及运算规则

绝大部分与APPLESOFT相同，主要增加如下内容：

整除

倒斜线“\”(即CTRL-B）表示整数除法，要求数值在－32768与32767之间，商只取整数部分，小数部分舍去。例如：

10 \ 4 =2

要注意的是整除的运算次序在乘法和浮点除法之后。

整数余数

整数余数以MOD表示，即取整数除法后整数的余数。例如：

11.369 MOD 3 = 2

(因为11 \ 3 = 3 余 2)

MOD的运算次序，位于整数除法之后。

除数为零或溢出时程序继续运行

MBASIC的程序在除数为零（或除数太小，接近于零）或计算溢出时，都会显示错误信息，但运算都继续进行，当然运算结果肯定是不正确的，例如下面这个程序与它运行的结果：

```
10 A=135.467
20 B=1E-38
30 C=A/B
40 LPRINT"A=";A
50 LPRINT"C=";C
60 D=0
70 E=A/D
80 LPRINT"A=";A
90 LPRINT"E=";E
```

```
A= 135.467
C= 1.70141E+38
A= 135.467
E= 1.70141E+38
```

程序运行中，首先屏幕显示OVERFLOW（溢出错），然后打印A值、C值；又显示DIVISION BY ZERO(除数为零错)，打印A值、E值。显然C值，E值都是不正确的。

3.6.5 MBASIC的命令和语句

以下按字母顺序排列介绍MBASIC的命令和语句。按惯例，命令指的是可以直接执行的指令；语句指的是在程序中执行的指令。有的指令既可作命令，也可作语句。介绍中凡与APPLESOFT功能相同的命令和语句从简，MBASIC与APPLESOFT指令的异同请见§ 3.9。

为了使用上的方便，以下把语法表达格式作如下规定：

〈 〉号内的内容必须由用户提供；

[ ] 号内的内容根据要求选用；

项目后的符号（…）表示可以重复使用该项目，直到该行的终点；

被垂直线符号（|）所隔开的两个项目，只能选择其中一项；

所有保留字的前后，都必须空一格；

除了角括号和中括号外，所有指令中标点符号不能任意取舍。

AUTO

是输入程序前预订自动输出行号的命令。AUTO的标准格式是AUTO [〈行号〉,[〈增量〉]]，有几种可能采用

的格式：

① AUTO

行号及增量都不写，则表示开始行号为10，增量为10.所输出的行号为10，20，30，…

② AUTO开始行号，增量

指定开始行号以及行号间隔。

如AUTO100,50输出行号为100,150,200,…

③ AUTO开始行号，

指定开始行号，行号间隔用上次AUTO指定的行号增量。

若键入AUTO命令之前已有行号相同的程序行，则在新行号之后先出现提示符*，这时若按下回车键，表示原先程序行保留，若打入新语句再键入回车，则原程序内容就被新语句内容冲掉。

键入CTRL-C即退出AUTO命令。

BEEP

为发声语句。

格式：

BEEP音高，维持时间

音高：0为最高音，255为最低音。

维持时间：0为最短时间，255为最长时间，约为1秒钟。

下面的程序产生一连串由高到低的声音。

```
10 FOR I=0 TO 255
20 BEEP I,I
30 NEXT
```

CALL

调用Z-80汇编语言或6502汇编语言的命令或语句。

语句格式1：

CALL〈变量名〉[(〈自变量表〉)]

目的：调用Z-80汇编语言子程序。

语句格式2：

CALL%〈变量名〉[(〈自变量〉)]

目的：调用6502汇编语言子程序。

其中变量各代表了子程序内存起始地址，但变量名不可以是数组变量名。自变量表指的是传送到汇编语言子程序的变量。

CHAIN

程序连接运行的语句，与APPLESOFT中用法不同。

语句格式：

CHAIN[MERGE]〈文件名〉[,[〈行号表达式〉][,ALL][,DELETE〈范围〉]]

① CHAIN文件名，行号，ALL

其中文件名指的是要求连接的磁盘上已知文件的名字；行号指的是要求连接文件的开始行号；ALL代表主机内存中正在运行程序的变量全部传送到要连接的文件中去执行。

例如磁盘上有已知程序文件L-12.BAS，如下：

```
100 LPRINT"A=";A
110 LPRINT"B=";B
120 FOR I=1 TO 11
130 LPRINT"C(";I;")=";C(I)
140 NEXT
```

现在在主机上键入以下程序：

```
10 DIM C(11)
20 FOR I=1 TO 11
30 C(I)=I
40 NEXT
50 A=111
60 B=222
80 CHAIN "B:L-12.BAS",100,ALL
```

第80行要求连接驱动器B:中磁盘文件L-12.BAS，并将其变量的值全部传送过来。运行这个程序的结果打印输出：

```
A= 111
B= 222
C( 1 )= 1
C( 2 )= 2
C( 3 )= 3
C( 4 )= 4
C( 5 )= 5
C( 6 )= 6
C( 7 )= 7
C( 8 )= 8
C( 9 )= 9
C( 10 )= 10
C( 11 )= 11
```

显然这是磁盘文件L-12.BAS引入主机运行的结果，所有变量值都已传送过来。这时执行LIST命令，屏幕显示的是所连接的L-12.BAS的程序。

② CHAIN MERGE文件名，行号，ALL

其中CHAIN MERGE表示要求连接和覆盖磁盘文件的命令。文件名指的是磁盘文件名字；行号指磁盘文件的开始行号；ALL指把全部变量传送到被连接的磁盘文件程序中去执行。这里有个规定，凡是被MERGE的磁盘文件要求原先是以ASCII格式存入的文件。

例如磁盘上有ASCII格式的文件QQQ．ASC如下：

```
100 LPRINT"A=";A
110 LPRINT"B=";B
120 FOR I=1 TO 11
130 LPRINT "C(";I;")=";C(I)
140 NEXT
```

现在在主机上键入以下程序：

```
1 DIM C(11)
2 FOR I=1 TO 11
3 C(I)=I
4 NEXT
5 A=111
6 B=222
7 CHAIN MERGE "B:QQQ.ASC",100,ALL
```

其中第7行要求连接运行和联上(作为覆盖)磁盘文件QQQ．ASC，并把全部变量传送过去执行。运行这个程序的结果是打印输出：

```
A= 111
B= 222
C( 1 )= 1
C( 2 )= 2
C( 3 )= 3
C( 4 )= 4
C( 5 )= 5
C( 6 )= 6
C( 7 )= 7
C( 8 )= 8
C( 9 )= 9
C( 10 )= 10
C( 11 )= 11
```

显然这已是磁盘文件QQQ.ASC引入主机运行后的结果，所有变量值都已传送过来。这时若执行LIST命令，则屏幕显示的是主机内原先程序与引入磁盘文件程序联在一起的程序如下：

```
1 DIM C(11)
2 FOR I=1 TO 11
3 C(I)=I
4 NEXT
5 A=111
6 B=222
7 CHAIN MERGE "B:QQQ.ASC",100,ALL
100 LPRINT"A=";A
110 LPRINT"B=";B
120 FOR I=1 TO 11
130 LPRINT"C(";I;")=";C(I)
140 NEXT
```

假如原先主机内程序有的行号与磁盘文件程序行号相同，则运行的结果将被顶替。

要注意的是，以上两种格式中，CHAIN格式不会保留被连接程序中的变量形态或者自定义的函数，必须在被连接程序中重新定义，而使用MERGE格式就可以保留。

CLEAR

语句格式：

CLEAR[,[〈表达式1〉] [,〈表达式2〉]]

将所有数值变量清除为零，字符串变量清除为空白。可用作命令或语句，表达式1指内存地址，表达式2指堆栈空间，用法与APPLESOFT类似。

CLOSE

语句格式：

CLOSE[#]〈缓冲区编号〉[,[#]〈缓冲区编号

…〉]

关闭文件，用法见 3.6.6 顺序处理文件及 3.6.7 随机存取文件。

COLOR

指定低分辨图形显示的颜色。

格式：

COLOR =〈颜色号码〉

有16种颜色：

0 黑色 1 紫红色 2 深蓝色 3 紫色 4 深绿色 5 灰色1 6 中等蓝色 7 淡蓝色 8 棕色 9 橙色 10 灰色2 11 粉红色 12 绿色 13 黄色 14 水色 15 白色

COMMON

语句格式：

COMMON 〈变量〉〈,变量〉…

在连接运行程序时，与CHAIN语句连用，把指定变量的值传送给被连接的程序使用，COMMON语句必须放在程序的最前面。同一个变量不能在一个以上的COMMON语句中出现，数组变量后面必须跟着空心的括号（ ）。例如，磁盘上有文件程序（L-12.BAS）如下：

```
100 LPRINT"A=";A
110 LPRINT"B=";B
120 FOR I=1 TO 11
130 LPRINT"C(";I;")=";C(I)
140 NEXT
```

现在设计一个程序，连接运行这个磁盘文件，并传送指定变量，如下：

```
10 DIM C(11)
20 FOR I=1 TO 11
30 C(I)=I
40 NEXT
50 A=111
60 B=222
70 COMMON A,C()
80 CHAIN "B:L-12.BAS"
```

第70行是传送变量A、数组C给被连接程序的命令，第80行是连接运行L-12.BAS的命令，命令中省去了行号和ALL。运行这程序时就接着把磁盘文件L-12.BAS引入主机运行，并打印输出：

```
A= 111
B= 0
C( 1 )= 1
C( 2 )= 2
C( 3 )= 3
C( 4 )= 4
C( 5 )= 5
C( 6 )= 6
C( 7 )= 7
C( 8 )= 8
C( 9 )= 9
C( 10 )= 10
C( 11 )= 11
```

显然这是把变量A和数组变量C传送过来的结果，变量B则没有传送过来。这时若键入LIST命令，显示出来的只是L-12.BAS的程序，原先主机中的程序已给冲掉了。

程序连接运行中，若需传送全部变量，其格式就是前面介绍过的CHAIN语句中使用参数ALL，这时就不必用COMMON语句了。

CONT

程序运行中，遇STOP或END语句停止运行时，可键入CONT命令使程序继续运行。程序运行中若键入CT-RL-C使程序暂停，也可键入CONT命令以继续运行程序。与APPLESOFT用法相同。

DATA

与READ语句配合使用，为置数语句，数据之间必须用逗号隔开。与APPLESOFT用法相同。

DEF FN

是用户自定义函数语句。与APPLESOFT用法稍不同，函数中可以有几个形式参数。

语句格式：

DEF FN<函数名>[(<形式参数><，形式参数>…)]=<函数表达式>

例如程序：

```
10 DEF FN ABC(X,Y,Z)=X^2+Y^2+Z^2+D^2
20 A=1:B=2:C=3
30 D=4
40 PRINT FN ABC(A,B,C)
```

运行这个程序的结果是屏幕显示数据30。由这个例子可知MBASIC中自定义函数语句特点：

① 形式参数可以用多个；

② 函数表达式中的变量可以用程序中运算结果的变量值。

DEF INT(SNG, DBL, STR)

语句格式：

DEF<类型><字母范围>

用于指定变量类型，其意义请参阅3.6.2。

DEF USR

语句格式：

DEF USR [〈数字〉]=〈整数表达式〉

指定汇编语言子程序的起始地址。

DEL

语句格式：

DEL[〈行号〉][—〈行号〉]

删除程序行的命令，也可用DELETE。

DEL 行号　删除指定行号的程序行。

DEL行号1—行号2　删除行号1与行号2之间所有程序行，包括这两个程序行。

DEL —行号　删除程序开始到指定程序行之间所有程序行，包括指定行号的程序行。

DIM

语句格式：

DIM〈数组变量〉〈,数组变量〉…

数组说明语句，与APPLESOFT用法类似。

EDIT

指定程序行进入编辑状态的命令。

格式：

EDIT 〈行号〉

这是MBASIC特有的命令，方便程序行的修改。键入EDIT命令后，指定的程序进入修改状态，先显示该行号，若按空格键，光标向右移，移过处的字符显示出来；若按I键，使光标所在位置处于插入字符状态，可以键入所需插入的字符，按ESC键则脱离插入状态，按回车键则直接输入整个

程序；若按D键，进入删除字符状态，删去光标右方一个字符，若按（数字）D键，则删去指定个数的字符，删去的字符显示出来，并用倒斜线区别出来；若按H键，删去光标右方的字符，自动进入插入状态；若按A键，回复这程序行原始的内容，光标移至程序行开头，以便重新修改这一行；若按X键，则光标移到程序末尾，这时可以继续键入字符。

在修改状态下，若键入命令不合法，则会发出警告声，不接受命令。

若要退出EDIT状态，只要按回车键，这时就把整个程序行都显示出来，并已进入内存。

上述编辑操作可以下表说明：

按键	功　能	光标动作
空格键	把移过处字符显示出来	右移
I 键	光标所在位置处于插入字符状态	不动
ESC键	脱离插入字符状态	不动
(数字)D键	删除光标所在位置右方指定个数的字符	删去的字符在一对倒斜线之间显示出来
H键	删去光标右方字符，自动进入插入状态	不动
A键	回复这程序行原始内容，以便重新修改	移到程序行开头
X键	继续键入程序内容	移到程序末尾
回车键	显示这程序行，进入内存	移入下一行

另一种情况是，当某程序运行时，若有错误，则自动进

入修改状态，修改完成之后，按回车键，这程序行又进入内存了，但这行原先变量的数值全部丢失了，再运行才能有；若程序修改后，按Q键，这时也会退出修改状态，但是程序实际上没有修改，而这行变量的值仍保留在内存中。

END

结束程序的运行，关闭所有文件。与APPLESOFT用法相同。

ERASE

语法格式：

ERASE〈数组变量〉〈，数组变量〉…

是取消程序中数组的命令，这时这个数组又可以重新用数组说明语句去说明了。例如下面这个程序，由于有了第20行的取消A数组的命令，第30行就可以重新定义数组了。

```
10 DIM A(13)
20 ERASE A
30 DIM A(11,11)
40 FOR I=1 TO 11
50 FOR J=1 TO 11
60 A(I,J)=I
70 PRINT A(I,J),
80 NEXT:NEXT
```

ERR和ERL

在处理错误流程时，ERR变量代表了错误代码，ERL变量代表了错误行的行号。这两个变量一般在条件语句中使用，指引程序分支。

ERROR

语法格式：

ERROR〈整数表达式〉

这个语句有两种用法：

① 若整数表达式的值正好对应MBASIC的某错误信息的代码，则会把错误信息印出，例如程序：

```
10 A=1
20 B=6
50 ERROR A+B
```

运行这个程序则显示代码7的内容，即：

OUT OF MEMORY IN 50

在直接命令方式中也可以类似方法使用，例如键入：

ERROR 7

则也会显示OUT OF MEMORY

② 这个命令也可使用户自行定义错误信息的代码，以指引程序的分支。整数表达式的值必须在0与255之间。例如以下程序及其运行结果：

```
10 ON ERROR GOTO 40
20 READ A
30 IF A>0 THEN ERROR 240 ELSE 60
40 IF ERR=240 THEN LPRINT "GOOD! A=";A
50 IF ERL=30 THEN RESUME 20
60 LPRINT "END"
70 DATA 24,369,-5

GOOD! A= 24
GOOD! A= 369
END
```

第10行指定程序运行出错时的分支，第30行自定义出错编码是240，第40行是条件分支语句，若错码等于240则打印，第50行指定若错码行的行号是30则重新读数据。运行的结果是把正值都打印出来，遇负值则打印END，运行结束。

FIELD

语法格式：

FIELD [#]〈缓冲区号码〉，〈字节长度〉AS 〈字符串变量〉…

用于指定随机文件缓冲区段划分。请见随机文件的介绍。

FILES

语法格式：

FILES [〈文件名〉]

表示把现在所选用驱动器中磁盘文件目录列出来。

例如：FILES "B：*.*"

指定把B驱动器中所有磁盘文件目录都列出。

当然也可指定列出某文件目录等。

FOR… NEXT

语法格式：

FOR〈变量〉=X TO Y [STEP Z]

⋮

NEXT [〈变量〉][,〈变量〉…]

式中X，Y，Z是数值表达式。

这是实行循环的语句，用法与APPLESOFT类似，但是在循环嵌套时，若循环在同一位置终止，NEXT可只写一个，后面跟着这几个循环变量即可〈循环变量间用逗号隔开）。

GET

有两种用法：

① 语法格式1：

GET[#]〈缓冲区号码〉[,〈记录号码〉]

用于随机存取文件，将记录读入缓冲区。请见随机存取文件介绍。

② 语法格式2：

GET〈键盘字母〉

这是从键盘上输入字母的语句，程序运行到这个语句时需要键盘输入字母，屏幕上则不显示，也不必按回车键。例如程序：

```
10 FOR I=1 TO 5
20 GET A$
30 LPRINT A$;
40 NEXT
```

运行这个程序到第20行就显示光标，等待键盘输入一个字母，例如5次循环分别输入ABCDE这5个字符，不必按回车键，就会输出字符串ABCDE，屏幕显示OK，表示运行终止。

GOSUB… RETURN

语法格式：

GOSUB〈行号〉

⋮

RETURN

为调用子程序及返回，用法与APPLESOFT语言相同。

GOTO

语法格式：

GOTO〈行号〉

为无条件转向语句，用法与APPLESOFT语言相同。

GR

语法格式：

GR [〈屏幕号码〉[,〈颜色编号〉]]

这是用于建立低分辨率图形模式的命令。屏幕号码为0至1，内容如下：

号码	屏幕状态
0	40×40的图形，4行文字说明
1	40×48的图形，没有文字说明

假如不指定屏幕号码，则自动确定号码为0。

语句中的颜色编号指定屏幕的背景颜色，若不指定颜色编号，则为黑色。

GR语法与APPLESOFT语言相同。

HLIN

语法格式：

HLIN 〈X_1的坐标〉，〈X_2的坐标〉 AT 〈Y的坐标〉（$0 \leqslant X_1$与$X_2 \leqslant 39$，$0 \leqslant Y \leqslant 47$）

这是低分辨图形模式中，从坐标(X_1, Y)至坐标(X_2, Y)画一条横线的语句。所画线的颜色由最近的COLOR语句指定。

HOME

语法格式：

HOME

为清除屏幕，使光标回到屏幕左上角的命令或语句。用法与APPLESOFT语言相同。

HTAB

语法格式:

HTAB 〈屏幕位置号码〉

指定光标移到屏幕位置号码所指定的位置，这个号码指的是绝对位置，而不是相对位移。屏幕上每行最左面号码是1，最右面号码是40；若指定的屏幕位置号码大于40，小于255，则根据这个数值被40除后的余数作为真正的屏幕位置号码；若指定的屏幕位置号码大于255，则显示错误信息:

ILLEGAL FUNCTION CALL（调用非法函数）。

IF…THEN…ELSE和IF…GOTO

有几种格式:

IF〈表达式〉THEN〈语句〉|〈行号〉[ELSE〈语句〉 〈行号〉]

IF〈表达式〉GOTO〈行号〉[ELSE〈语句〉|〈行号〉]

在APPLESOFT语言中没有上述的ELSE语句，即假如表达式结果为零，则执行ELSE后面的语句（或行号），其它的用法两者类似。

INPUT

程序运行中，执行键盘输入的语句.

语法格式有几种:

① INPUT〈变量〉，〈变量〉，…

程序运行到这个语句，出现问号，要求键盘输入资料，变量之间必须用逗号隔开。

② INPUT〈"字符串"〉；〈变量〉，〈变量〉，…

程序运行到这个语句，显示字符串，后面跟问号，要求键盘输入资料，变量之间必须用逗号隔开。

③ INPUT 〈“字符串”〉, 〈变量〉, 〈变量〉, …

程序运行到这个语句，显示字符串，不跟问号，要求键盘输入资料。

例如下面这个程序：

```
10 INPUT "A,B=";A,B
20 INPUT "C=",C
30 LPRINT "A=";A
40 LPRINT "B=";B
50 LPRINT "C=";C
```

运行这个程序到第10行，显示A，B=？要求键盘输入，若键入1，2则继续执行到第20行，显示C=要求键盘输入，若键入3则顺利运行打印输出：

```
A= 1
B= 2
C= 3
```

假如键盘输入资料太多或太少，屏幕将显示？Redo from start要求重新输入，这一点与APPLESOFT语言稍有不同。

INPUT

语法格式：

INPUT# 〈缓冲区号码〉, 〈变量〉, …

从顺序处理磁盘文件输入资料的语句。详见顺序处理文件和随机存取文件的介绍。

INVERSE

语法格式：

INVERSE

为使显示屏成为白底黑字画面的命令或语句。

KILL

语法格式：

KILL 〈文件名〉

用于删除所有类型磁盘文件的命令或语句，但是不能删除目前已打开的文件。

LET

语法：[LET]〈变量〉=〈表达式〉

这是赋值语句，用法与APPLESOFT语言相同，也可以省略。

LINE INPUT

语法格式：

LINE INPUT〈字符串变量〉

LINE INPUT〈"提示字符串"〉；〈字符串变量〉

这是允许整行输入字符串变量的语句，字长不超过254个字符。程序运行中遇到这种语句，会出现光标，若用了字符串，则会显示提示字符串，要求键盘输入字符串，若按CTRL-C键，则暂停执行LINE INPUT，屏幕显示"OK"，键入CONT，则继续执行。以下是个例子，并与INPUT语句比较：

```
10 INPUT "A$=",A$
20 LPRINT "A$=";A$
100 LINE INPUT "B$=";B$
110 LPRINT "B$=";B$
```

运行这个程序到第10行显示A$ =，键入QWE回车，则运行到100行显示B$ =?，若键入QWER，1234，回车，则程序执行完毕，打印输出：

```
A$=QWE
B$=QWER,1234
```

LINE INPUT#

语法格式:

LINE INPUT# 〈缓冲区号码〉,〈字符串变量〉

用于以整行方式,从顺序处理文件中读取资料给指定的字符串变量。也可用于随机文件中,在GET语句之后,从随机文件缓冲区读入资料。详见顺序处理文件和随机存取文件的介绍。

LIST

用于列出程序。有多种格式:

LIST

LIST 行号

LIST ,行号

LIST 行号,

LIST 行号,行号

用法与APPLESOFT语言相似,逗号也可用负号代替。

LLIST

用于程序的打印输出,对CP/M2.2版本,要求先执行DDT命令(见前面的介绍),语法与LIST类似。

LOAD

语法格式:

LOAD〈文件名〉[,R]

用于把磁盘文件引入内存。执行这命令后,关闭目前在内存中的文件,删去所有现用变量值。命令最后面的参数R表示磁盘文件中程序引入内存后,立即自动运行。

LPRINT和LPRINT USING

用于打印机上输出资料的命令或语句,有几种格式:

① LPRINT

打印一空白行

② LPRINT〈表达式〉

打印输出表达式，最多可打印输出132个字符。

③ LPRINT USING 〈字符串表达式〉；〈表达式〉

这种形式的语句指定表达式的输出格式，其中表达式可以是字符串或数值表达式。格式中的字符串表达式指定了输出的格式：

a）输出对象是字符串时

这时语句中的表达式当然是字符串。字符串表达式有三种，先看下面这个程序和运行的结果：

```
10 A$="ABCD":B$="1234567"
20 LPRINT USING "!";A$;B$
30 LPRINT USING "öö";A$;B$
40 LPRINT USING "ö  ö";A$;B$
50 LPRINT USING "ö    ö";A$;B$
60 LPRINT USING "&";A$;B$

A1
AB12
ABCD1234
ABCD  123456
ABCD1234567
```

第20行用符号“！”指定只输出字符串的第一个字符；
第30行用“\\ ”指定输出字符串的 2＋n 个字符[1]，在这程序行中n＝0（指两斜杠之间无空格），故只印出字符串的 2 个字符，第40行n＝2，故印出 4 个字符，第50行n＝4，指定印出 6 个字符，故A＄后面加了两个空格输出；第60行

1）有的打印机把符号 \ 打成 ö。

以“&”指定原样输出字符串，这个语句与打印一个空白行的LPRINT语句相同。

b）输出对象是数据时

这时语句中的表达式是数据表达式，字符串表达式有许多种形式，看下面这个程序：

```
10 A#=123456.789#:B=-123.45
20 LPRINT USING "###.##";A#,B
30 LPRINT USING "#######.####";A#,B
40 LPRINT USING "+#######.##";A#,B
50 LPRINT USING "**######.##";A#,B
60 LPRINT USING "$$######.##";A#,B
70 LPRINT USING "**$#####.##";A#,B
80 LPRINT USING "**$#####,.##";A#,B
90 LPRINT USING "##.##^^^^";A#,B
```

运行输出结果：

```
%123456.79%-123.45
 123456.7890    -123.4500
 +123456.79     -123.45
*123456.79***-123.45
$123456.79   -$123.45
$123456.79**-$123.45
$123,456.79***-$123.45
 1.23D+05-1.23E+02
```

第20程序行以“#”的形式指定数据输出格式，#的位数决定输出数据的位数，若指定位数不够用，则输出数据前面印出符号%，小数则按指定的位数四舍五入输出；第30程序行类似，指定的小数位多时，输出补零；第40程序行增加了符号“+”，使正数输出时有正号；第50程序行以符号“**”指定当预定位数多时，输出数据前面标以*号；第60程序行以

符号“$ $”指定输出数据前面标以$号；第70程序行以符号“* * $”指定当预定位数多时，输出数据前面标以*号；第80程序行以符号“,”号指定输出数据每3位印出一个逗号；第90程序行以“↑↑↑↑”指定以指数形式输出数据。

规定指定的数值位数不能超过24。

LSET和RSET

语法格式：

LSET <字符串变量>= <字符串表达式>

RSET <字符串变量>= <字符串表达式>

用于将资料从内存中移至随机文件缓冲区中。详见随机存取文件的介绍。

MERGE

语法格式：

MERGE <文件名>

引进指定的磁盘文件，与内存中原有程序连接的命令。要求这个磁盘文件原先是用ASCII格式贮存的。若磁盘文件中行号与内存中程序行号相同，则引入程序的行号会取代内存原有程序的行号。MERGE命令使用例如下。磁盘文件QQQ.ASC的程序为：

```
100 LPRINT"A=";A
110 LPRINT"B=";B
120 FOR I=1 TO 11
130 LPRINT "C(";I;")=";C(I)
140 NEXT
```

这时若在主机内存中键入如下程序：

```
10 DIM C(11)
20 FOR I=1 TO 1
```

```
30 C(I)=I
40 NEXT
50 A=111
100 B=222
```

又键入直接命令MERGE "QQQ.ASC",则会把磁盘文件引入主机，并已与主机内原有程序相连接，执行LIST命令，列表如下：

```
10 DIM C(11)
20 FOR I=1 TO 11
30 C(I)=I
40 NEXT
50 A=111
100 LPRINT"A=";A
110 LPRINT"B=";B
120 FOR I=1 TO 11
130 LPRINT"C(";I;")=";C(I)
140 NEXT
```

MID$

用另一字符串代替原有字符串的一部分，有两种用法：

① MID$(〈字符串表达式1〉，N)=〈字符串表达式2〉

表示对于字符串表达式1，从第N个字符开始，用字符串表达式2全部字符来替换，要求替换的字符长度不可多于原字符串表达式，否则截去多余部分。

② MID$(〈字符串表达式1〉，N，M)=〈字符串表达式2〉

表示对于字符串表达式1，从第N个字符开始，用字符串表达式2的M个字符替换。

例如以下程序：

```
10 A$="1234567.99999"
20 MID$(A$,9)="123"
30 LPRINT"A$=";A$
40 MID$(A$,9,2)="45678"
50-LPRINT"A$=";A$
```

运行则打印输出:

```
A$=1234567.12399
A$=1234567.45399
```

第20行把原先的A$替换了三个字符;第40行把已改变了的A$又替换了二个字符。

NAME

语法格式:

NAME 〈旧文件名〉AS 〈新文件名〉

用于更换磁盘文件名的命令。

NEW

清除内存中全部程序和变量,与APPLESOFT语言用法相同。

NORMAL

将屏幕转变为黑底白字画面的命令,常与INVERSE连用。

ON ERROR GOTO

语法格式:

ON ERROR GOTO〈行号〉

用于指定程序运行中出错时处理错误的流程的语句。

ON…GOSUB和ON…GOTO

语法格式:

ON 〈表达式〉 GOTO 〈行号1〉,〈行号2〉,…

ON 〈表达式〉GOSUB 〈行号1〉,〈行号2〉,…

(ON…GOSUB 后的行号，必须是转向程序的第一个语句的行号)

这是开关型语句，根据表达式的值（1,2,…）决定程序转向的行号，若表达式的值为零或大于行号群的个数，则这语句不执行，执行下一句；若表达式的值大于255或小于零，则出错，显示Illegal function call（调用非法函数）。

OPEN

语法格式：

OPEN〈"状态"〉,［#］〈缓冲区号码〉,〈文件名〉［,〈记录长度〉］

用于顺序处理文件和随机存取文件中打开文件用的语句。详见顺序处理文件和随机存取文件的介绍。

OPTION BASE

语法格式：

OPTION BASE n

式中 n 为 0 或 1 。指定数组下标的最小值，若不写 n ，表示最小下标为零。

PLOT

语法格式：

PLOT〈X坐标〉，〈Y坐标〉

式中X，Y均为整数表达式，0≤X≤39，0≤Y≤47。这个语句用于低分辨率图形显示，以X，Y坐标定点。

POKE

语法格式：

POKE I, J

式中 I，J 都是整数表达式。0≤I≤65536，0≤J≤255

这个语句把内容J写进主机内存地址I处。

POP

语法格式：

POP

这个语句用于GOSUB之后，无法返回到GOSUB的下一个语句时，用它使流程从子程序中返回。例如下面的程序和输出结果：

```
10 LPRINT"EXAMPLE:"
20 GOSUB 100
30 LPRINT"ABC"
40 END
100 GOSUB 200
110 LPRINT"CDE"
120 RETURN
200 POP
210 RETURN

EXAMPLE:
ABC
```

PRINT

用于屏幕显示资料，有多种使用格式，与APPLESOFT语言用法类似。

PRINT USING

语法格式：

PRINT USING〈字符串表达式〉；〈表达式〉

用于以指定格式显示数据或字符串，用法与LPRINT USING类似，可以参照使用。

PRINT # 和 PRINT # USING

语法格式：

PRINT#〈缓冲区号码〉，〈表达式〉

PRINT＃〈缓冲区号码〉，USING〈字符串表达式〉；〈表达式〉

详见顺序处理文件和随机存取文件的介绍。

PUT

语法格式：

PUT[＃]〈缓冲区号码〉[,〈记录号码〉]

是用于把记录从随机文件缓冲区写入随机文件的语句。详见随机存取文件的介绍。

RANDOMIZE

语法格式：

RANDOMIZE[〈表达式〉]

用于产生随机数的语句，若不写表达式，则程序的运行暂停，要求键盘输入，并做出如下显示：

Random Number Seed (－32768 to 32767)?

假如键入的值相同，则运用RND函数时所得结果也相同。例如下面的程序：

```
10 FOR J=1 TO 3
20 RANDOMIZE J
30 FOR I=1 TO 4
40 LPRINT RND,
50 NEXT
60 LPRINT
70 NEXT
```

运行后输出：

```
.58041     .128928    .928324    .901162
.89113     .454173    .881064    .438379
.168836    .784757    .665827    .463432
```

READ

语法格式：

READ 〈变量〉，〈变量〉，…

用于读入资料的语句，必须与DATA语句配合使用。与APPLESOFT语言用法相同。

REM

语法格式：

REM 〈说明〉

在机器内并不执行操作的解释性语句，用法与APPLESOFT 语言相同。另外亦可把说明放在单引号中，附在程序行的后面，例如：

```
10 FOR I=1 TO 5 'THIS IS A STATEMENT'
20 PRINT I;
30 NEXT
```

RENUM

改变程序行号的命令。语法格式：

RENUM[[〈新号码〉][,[〈归号码〉][,〈增量〉]]]

① RENUM

把程序行号改变为以10开始，增量为10的新行号。

② RENUM 行号X

把程序行号改变为以行号X开始，增量为10的新行号。

③ RENUM 行号X，，增量Z

把程序行号改变为以行号X为开始行，以Z为增量的新行号。

④ RENUM行号X，行号Y，增量Z

把原程序从行号Y开始，改变为行号X开始，增量为Z的新行号。

RENUM命令不能改变程序中各行的次序，行号不能

超过65529，否则显示出错信息。

RESET

语法格式：

RESET

用于驱动器中的换磁盘之后，重新分配目录资料。若是换了磁盘后，不键入这个命令，则当执行资料存入磁盘等操作时，会显示出错信息：DISK READ ONLY（磁盘只读）。

换磁盘的正常步骤应该是：

a) 键入CLOSE命令，以关闭所有文件；

b) 换入新磁盘；

c) 键入RESET命令。

RESTORE

语法格式：

RESTORE[〈行号〉]

用于数据恢复的语句，用法与APPLESOFT语言相同。

RESUME

用于检查出错误后，继续执行程序的语句。有多种语法格式：

① RESUME或RESUME 0

在发生错误的地方恢复执行程序。

② RESUME NEXT

在发生错误的下一行恢复执行程序。

③ RESUME〈行号〉

在指定的行号处恢复执行程序。

例如以下程序：

```
10 ON ERROR GOTO 100
20 A=123
30 C=A/B
40 PRINT"A=";A
100 RESUME NEXT
```

运行显示：A＝123

这语句与APPLESOFT语言用法不同。

RUN

这是运行程序的命令，有多种语法格式：

① RUN

从头开始运行程序。

② RUN 〈行号〉

从指定行号开始运行程序。

③ RUN "磁盘文件名"

先关闭所有打开的文件，然后从磁盘调入指定文件，立即自动运行。

④ RUN "磁盘文件名",R

调入指定的磁盘文件运行，所有打开的资料文件仍再打开。

SAVE

将程序存入磁盘的命令，有三种语法格式：

① SAVE "文件名"

将指定文件存入磁盘，存入的是二进制格式。

② SAVE "文件名",A

将指定文件以ASCII格式存入磁盘，这种存入格式占空间较大，但可用于MERGE及LIST命令等。

③ SAVE“文件名”,P

将指定文件以二进制格式存入磁盘，并进行保护。这时可以调入这个文件，或用运行文件命令把它从磁盘调入立即运行，但是只能显示运行结果，无法修改或用LIST命令列表。

STOP

语法格式：

STOP

用于暂停执行程序，回至OK状态，若键入CONT则继续执行。与APPLESOFT语言用法相同。

SWAP

语法格式：

SWAP 〈变量〉，〈变量〉

用于将两个变量的值互相交换的语句。例如程序：

```
10 A=111:B=999
20 LPRINT"A=";A
30 LPRINT"B=";B
40 LPRINT
50 SWAP A,B
60 LPRINT"A=";A
70 LPRINT"B=";B
```

运行后输出：

```
A= 111
B= 999

A= 999
B= 111
```

SYSTEM

语法格式：

SYSTEM

用于关闭所有文件，回至CP/M控制下的命令。

TEXT

语法格式：

TEXT

用于从图形显示回到文本显示（24行，40列），从低分辨图形返回TEXT时，能清除画面。

TRACE/NOTRACE

语法格式：

TRACE

NOTRACE

TRACE为进入跟踪，NOTRACE为消除TRACE，退出跟踪，可以用命令或语句形式，用于程序调试。NEW可以代替NOTRACE。

VLIN

语法格式：

VLIN 〈Y1坐标〉，〈Y2坐标〉AT 〈X坐标〉

式中 $0 \leqslant Y1 \leqslant 47$， $0 \leqslant Y2 \leqslant 47$， $0 \leqslant X \leqslant 39$

用于低分辨图形显示时，从点（X，Y1）至点(X，Y2)画一条垂线，Y1必须≤Y2，颜色由刚才执行的COLOR语句决定，在TEXT状态或图形、文字混合状态，画出的是字符组成的线。

例如画一条垂线的程序：

```
10 GR
20 COLOR=3
30 VLIN 1,47  AT 20
```

VTAB

语法格式：

VTAB 〈屏幕行号〉

用于将光标移到指定的屏幕行的命令或语句，屏幕共24行，例如 VTAB 5 指将光标移到第5行；若屏幕行号超过24，则将该数除24后所得余数即为指定的行，如 VTAB 30 则指定光标移到第6行；若屏幕行号超过255，则显示出错信息：

ILLEGAL FUNCTION CALL

WAIT

语法格式：

WAIT 〈地址〉，I [，J]

其中I，J 是整数表达式，若不写J，则表示J 值为零。

这个语句用于检查地址状态时，程序暂停运行。

WHILE…WEND

语法格式：

```
WHILE 〈表达式〉
      ⋮
[〈循环语句〉]
      ⋮
  WEND
```

这是控制是否执行循环的语句，表达式不为零时，则执行循环，然后从 WEND 语句回到WHTLE，再用表达式判断；若表达式为零，则执行WEND后面的语句，WHILE…WEND 可以嵌套使用。例如程序：

```
10 A=3
20 WHILE A
```

```
30 FOR I=1 TO 5
40 LPRINT I;
50 NEXT
55 LPRINT
60 A=A-1
70 WEND
80 LPRINT"JUMP"
```

运行则输出：

```
 1   2   3   4   5
 1   2   3   4   5
 1   2   3   4   5
JUMP
```

WIDTH

用于指定屏幕显示行宽及高度或打印机输出的行宽的命令或语句，格式如下：

① WIDTH [〈行宽〉]，[〈屏幕高度〉]

指定屏幕显示行宽及高度。15≤行宽≤255，1≤屏幕高度≤24，对APPLE屏幕，语句格式中省略行宽，则行宽自动定为40，省略屏幕高度，则屏幕高度自动定为24。

② WIDTH LPRINT〈行宽〉

指定打印输出的行宽。15≤行宽≤255

WRITE

语法格式：

WRITE[〈表达式〉]

用于在终端输出资料的命令或语句。

WRITE 显示一个空行。

WRITE〈表达式〉，〈表达式〉，… 将把各表达式的值显示出来，中间自动用逗号隔开，字符串用引号括出。

侧如以下程序：

```
10 A=3:B=6:C$="A"
20 WRITE A.B.C$
```

运行显示：3,6，“A”

WRITE #

语法格式：

WRITE # 〈缓冲区号码〉，〈表达式〉，〈表达式〉，…

详见顺序处理文件和随机存取文件的介绍。

3.6.6 顺序处理文件

建立缓冲区

MBASIC程序中控制资料在磁盘上的存取方法与APPLESOFT不同。首先要设置输入输出缓冲区，也即等待区的意思，所有进出磁盘文件的资料都必须经过这个缓冲区，一般说来，程序中同时使用多少个文件，就需要建立多少个缓冲区。在前面的3.6.1中介绍的进入MBASIC的工作方式，就有指定允许同时使用多少个文件，也即建立多少个缓冲区的命令，如：

A〉MBASIC 〈CR〉

这时进入的MBASIC操作方式，指定建立3个缓冲区。

A〉MBASIC/F：文件数目

这时进入的MBASIC操作方式，指定建立了所标“文件数目”的缓冲区。其中文件数目可取1与15之间的一个整数。每个缓冲区在内存中需占200多个字节，建立的缓冲区数目越多，所占内存空间也越大，因此必须根据具体情况适当指定多少个缓冲区。

顺序处理文件方式存取资料

顺序处理文件存取资料方式的概念与APPLESOFT类似，即是按资料的排列次序顺序存取，但程序编制是不同的。

① 打开文件语句

存取文件之前先要打开文件，格式是：

OPEN“存取类型”,[#]缓冲区编号，“文件名称”

其中“存取类型”指定了存取类型,“O”为存,“I ”为取(都指顺序处理文件)；缓冲区编号指定了文件所用缓冲区，“文件名称”指这个顺序处理文件的名字。例如：

10 OPEN “O”，2，“ABC. TXT”

指为了建立顺序处理文件，使用2号缓冲区，打开文件ABC. TXT，若原先这个文件已存在,则把原先内容删除。

关于OPEN语句，有几点要注意：

a）缓冲区被OPFN语句打开后，就不能再在别的OPEN语句中使用，须先关闭（CLOSE）文件，这个缓冲区才能另作它用。

b）顺序输入(I)的一个文件可以设置多个缓冲区,但顺序输出(O)的每个文件只能设置1个缓冲区。

② 关闭文件语句

被打开文件用完之后必须关闭，否则可能会丢失资料。关闭文件的语句格式：

CLOSE缓冲区编号，缓冲区编号，…

若不指定缓冲区编号；则为关闭所有文件。另外在执行了NEW，RUN，MERGE，CLEAR等命令，或删改了文件中的某个程序行，或增删了某些程序，或磁盘错误(磁盘已满的错误除外），都会自动关闭所有文件。

③ PRINT# 语句

这是把资料写入文件的语句，格式是：

PRINT#缓冲区编号，表达式；表达式…

表达式值计算结果，能顺序写入指定缓冲区的文件内。

有几点注意：

a）用PRINT #语句写入的数据是ASCII 代码格式，在CP/M操作方式（A>）下可用直接命令 TYPE 调出显示。而且在磁盘里建立的信息格式与 PRINT 语句在显示屏上输出格式相同，所以为了节省磁盘空间，表达式之间最好用分号隔开。

b）若是把几个字符串写入文件，要求字符串变量之间用分隔符逗号或分号隔开。若字符串中包含逗号，则在 PRINT# 语句中应该用引号CHR$（34），把字符串两边括起来，例如：

PRINT#1，CHR$（34）；A$；CHR$（34）；B$

c）当所有 PRINT # 语句执行完毕时，要使用CLOSE 语句来关闭文件。

④ INPUT#语句

这是从磁盘文件顺序读进资料的语句，格式是：

INPUT#缓冲区编号，变量，变量……

以下用一些例子来说明上述语句的使用方法：

例一，建立顺序处理文件

```
10 A$="ABCD"
20 B$="1234"
30 OPEN "O",#1,"DATA-O"
40 PRINT#1,A$;",";B$
50 CLOSE
```

这个程序把两个字符串通过缓冲区1号存入顺序处理文件DATA-0，字符串变量间用分号隔开，能节约所占磁盘空间，中间的带引号的逗号是分隔两个字符串必不可少的。

例二，读取顺序处理文件

```
10 OPEN "I",#2,"B:DATA-0"
20 INPUT #2,C$,D$
30 CLOSE
40 LPRINT C$,D$

ABCD             1234
```

这个程序通过缓冲区2号，把顺序处理文件DATA-0中的内容读取出来（程序下面是运行结果）。

例三，资料在磁盘上存贮格式

```
10 A$="ABCD"
20 B$="1234"
30 OPEN "O",#1,"B:DATA-1"
40 PRINT#1,A$,B$
50 CLOSE
```

第40行存 A$ 与 B$ 之间没有加分隔符“,”,在磁盘上这两个字符串连接在一起了。

然后用下面的程序从顺序处理文件DATA-1中把存入的字符串取出来：

```
10 OPEN "I",2,"B:DATA-1"
20 INPUT#2,C$
30 CLOSE
40 LPRINT C$

ABCD             1234
```

第20行只用一个字符串变量 C$ 即把 DATA-1 中的内

容取出来了，事实上，若是用两个变量取资料，就会因资料不够而出错，因为在存字符串的时候，这两个字符串已合并为一个了，又因为存入时变量之间又用逗号分隔，(注意不能用分号，字符串将完全连在一起，中间没有空格）所以合并的字符串间仍有许多空格。

例四，包含有逗号等字符的字符串存入顺序处理文件

```
10 A$="123,456"
20 OPEN "O",1,"B:DATA-2"
30 PRINT#1,CHR$(34);A$;CHR$(34)
40 CLOSE
```

前面已经说过CHR$(34)是ASCII代码形式的双引号，第30行把带逗号（或分号或空格等）的字符串存入顺序处理文件DATA-2时，两面用CHR$(34)括了起来。下面这个程序又从DATA-2中取出了存入字符串：

```
10 OPEN "I",2,"B:DATA-2"
20 INPUT#2,C$
30 CLOSE
40 PRINT C$
```

运行输出：

123,456

例五，LINE INPUT #的用法

```
10 OPEN "I",2,"B:DATA-0"
20 LINE INPUT#2,C$
30 CLOSE
40 LPRINT C$
```

在例一中建立了DATA-0文件，包含两个字符串，上面程序中用LINE INPUT #语句从DATA-0中取内

容，运行输出：

ABCD，1234

可知 LINE INPUT＃ 语句与LINE INPUT语句类似，是把整个可输入字符串全部输入。

例六，数据存取

```
10 READ N%
20 OPEN "O",#1,"B:DATA-3"
30 FOR I%=1 TO N%
40 READ X,Y
50 PRINT#1,X,Y
60 NEXT
70 CLOSE
100 DATA 10
110 DATA 1.1,5.1,1.2,5.2,1.3,5.3,1.4,5.4,1.5,5.5
120 DATA 1.6,5.6,1.7,5.7,1.8,5.8,1.9,5.9,2.1,6.1
```

这个程序建立了顺序处理文件 DATA-3，其中存20个数据。下面程序从 DATA-3中取出数据：

```
10 READ N%
20 OPEN "I",#1,"B:DATA-3"
30 FOR I%=1 TO N%
40 INPUT #1,A,B
50 LPRINT A,B
60 NEXT
70 LPRINT LOC(1)
80 CLOSE
90 DATA 10
```

```
1.1          5.1
1.2          5.2
1.3          5.3
1.4          5.4
1.5          5.5
1.6          5.6
1.7          5.7
```

```
1.8            5.8
1.9            5.9
2.1            6.1
2
```

其中70行用到了LOC（缓冲区号码）函数，即要求打开文件后打印输出读出的扇区数（这里扇区数=2），类似在建立顺序处理文件时找出写入的扇区数。

例七，同时建立几个文件

为要同时建立几个文件，就必须打开几个缓冲区。

```
10 READ N%
20 OPEN "O",#1,"B:DATA-5"
30 OPEN "O",#2,"B:DATA-6"
40 FOR I%=1 TO N%
50 READ X,Y
60 PRINT#1,X
70 PRINT#2,Y
80 NEXT
90 CLOSE 1,2
100 DATA 10
110 DATA 1.1,5.1,1.2,5.2,1.3,5.3,1.4,5.4,1.5,5.5
120 DATA 1.6,5.6,1.7,5.7,1.8,5.8,1.9,5.9,2.1,6.1
```

通过这个程序，把X与Y分别存入两个不同的文件。

另外亦可用WRITE #代替PRINT #把资料存入顺序处理文件，格式是：

WRITE #缓冲区号码，表达式，表达式，…

它的特点是写入磁盘的资料之间会自动加入逗号；若资料是字符串，则会写入引号。

在顺序处理文件中增加资料

由于顺序处理文件中存取资料是按次序进行的，存资料用型号“O”,取资料用“I”,类型不同，所以在文件中增加资料就比较麻烦，步骤是：

a) 在“I”状态下打开原文件；

b) 在“O”状态下打开一个过渡文件；

c) 从原文件读资料，写入过渡文件；

d) 关闭原文件，并删除；

e) 把增加的资料写入过渡文件；

f) 关闭过渡文件，并把过渡文件换名为原文件名。

这时这个文件中已增加了资料。

例如下面的程序建立了顺序处理文件DATA-4。

```
10 OPEN "O",2,"B:DATA-4"
20 FOR I=1 TO 5
30 READ X
40 PRINT#2,X
50 NEXT
60 CLOSE
100 DATA 34,23,86,-21,0
```

现在DATA-4中有5个数据，要求再增加一批数据。下面这个程序（例八）就是在DATA-4中增加资料的程序：

```
10 OPEN "I",1,"B:DATA-4"
20 OPEN "O",2,"B:COPY"
30 IF EOF(1) THEN 70
40 INPUT#1,X
50 PRINT#2,X
60 GOTO 30
70 CLOSE 1
80 KILL "B:DATA-4"
90 FOR I=1 TO 6
100 READ X
110 PRINT#2,USING "$$#####.##";X
120 NEXT
130 CLOSE
140 NAME "B:COPY" AS "B:DATA-4"
300 DATA -1266.56,44324.11,-2897.67,33,-22.1,67.745
```

第10行打开原文件为了读；第20行建立过渡文件COPY；第30行使用EOF（缓冲区编号）函数，这个函数能够检查文件是否结束，若文件已结束，函数取真值，否则取假值，有了这个函数，这个程序就能顺利执行；第70行只关闭原文件；第80行删除原文件；第90—120行在过渡文件中增加资料，其中第110行用了格式语句，与PRINT USING用法类似；第140行把过渡文件名字改为原文件名，达到了在原文件中增加资料的目的。

现在要检查这个文件中是否确实增加了资料，当然可以编一个程序来做到这一点，但是更简单的办法是键入SYSTEM命令，使回到提示符A〉，再键入TYPE命令，由于顺序处理文件中资料是按ASCII格式存贮的，这时就会把所有资料输出，如下：

```
 34
 23
 86
-21
 0
-$1266.56
$44324.10
-$2897.67
    $33.00
   -$22.10
    $67.75
```

3.6.7 随机存取文件

特点

随机存取文件的存取格式与APPLESOFT语言类似，主要是：

① 随机存取文件中按文件记录号存取资料，存取速度快，修改文件和存取个别数据灵活方便。

② 随机文件以密集的二进制格式存贮资料，所占磁盘容量较少。

建立随机存取文件的程序格式

例九，把字符串存入随机文件

```
10 A$="ABCDE":B$="123"
20 OPEN "R",#1,"B:DATA-1.TXT"
30 FIELD#1,6 AS X$,5 AS Y$
40 LSET X$=A$
50 LSET Y$=B$
60 PUT 1,2
70 CLOSE 1
```

第20行是打开文件的命令。建立随机文件之前也必须先打开文件，类型用“R”，缓冲区编号任选，后面指出文件名，还可跟记录长度项，故打开文件的格式是：

OPEN“R”，[#]缓冲区编号，“文件名”，记录长度

由前面 3.6.1 中介绍可知，在进入MBASIC工作方式时已预定了随机文件的记录长度，如：

A〉MBASIC〈CR〉

即进入MBASIC工作方式，指定的随机文件的每个记录长度是128个字节。

A〉MBASIC /S：记录的最大长度〈CR〉

这时进入MBASIC 工作方式，指定了随机文件每个记录的最大长度。

因此在随机文件打开时（OPEN语句），所指记录长度不能超过进入MBASIC 工作方式时已指定的记录长度，若OPEN语句中不写明记录长度，这时指定的记录长度就是进入MBASIC工作方式时已确定的记录长度。

缓冲区编号的用法与顺序处理文件一样，前面可标符号#，也可不标。

第30行FIELD语句把缓冲区划分为区段，以便存放资料，例如这个程序中，开头6个字节命名为X$，随后5个字节命名为Y$，剩下的其它字节没有用到。FIELD语句的格式是：

FIELD缓冲区编号，长度1 AS区段名1，…

关于FIELD语句要注意：

① 可以多次使用FIELD语句，以便在缓冲区的同一区域中，组织不同的区段，例如：

30 FIELD1，60 AS X$

40 FIELD1，50 AS A$，10 AS B$

表示1号缓冲区，X$区段为60个字节，但可组织两个区段，A$（50字节）和B$（10字节）。

② 区段名只代表FIELD语句设置的缓冲区区段，不再具有"字符串变量"含义，若在以下程序的赋值语句左面重复使用缓冲区区段名，则它就不再是缓冲区区段名，FIELD语句即失效。

第40行和50行的LSET语句把字符串放入缓冲区区段里。LSET语句的格式：

LSET区段名＝字符串数据

LSET语句是把字符串数据从左面开始放到缓冲区区段中去的，若字符串长度小于区段长度，则区段里右面填满空格；若大于区段长度，则将右面多余的字符去掉。

另有RSET语句，也可把字符串放入缓冲区的区段中，格式是：

RSET区段名＝字符串数据

RSET语句把字符串数据放入缓冲区的区段的格式与LS-ET稍不同，当字符串长度小于区段长度时，在左边填满空

格。

当把数值存入随机文件时，先要把数值转换为字符串才行，对于不同类型的数值有不同的转换函数：

MKD$（数表达式）函数把双精度的值转换成8个字节的字符串。

MKS$（数表达式）函数把单精度值转换为4个字节的字符串。

MKI $（数表达式）函数把整型值转换为2个字节的字符串。

第60行PUT语句把指定缓冲区的内容写入磁盘文件，并指定记录号，格式是：

PUT缓冲区编号，记录号

若是语句中省略了记录号，则使用的是当前的记录号，第一次存取文件时，当前的记录编号为1，以后每执行一次省略记录号的PUT语句（或GET语句）都使当前的记录编号自动加1。

第70行为关闭文件语句，用法与顺序处理文件相同。

取随机存取文件的程序格式

例十，从随机文件中取出字符串

```
10 OPEN "R",#2,"B:DATA-1.TXT"
20 FIELD #2,6 AS M$,5 AS N$
30 GET #2,2
40 LPRINT M$,N$
50 CLOSE
```

```
ABCDE          123
```

第10行和20行与建立随机文件时相同；第30行用GET

语句从磁盘文件中读出一个指定记录，把其内容放入指定的缓冲区里，格式是：

GET缓冲区编号，记录号

GET语句中各参数规定及记录号用法与PUT语句相同。

若是要把某随机文件中存的数据取出来，则必须使用字符串转换为数值的函数（例十中没有要求取出数据，123仍旧是字符串）。

CVD（区段名）函数能把指定区段里的8个字节的字符串转换成双精度数值。指定区段里的字符串的长度必须等于8，若小于8，则出错；若大于8，则只取前8个字符。函数CVD与MKD $函数的转换过程正好相反。

CVS（区段名）函数把指定区段里的4个字节的字符串转换成单精度值。指定区段里的字符串的长度必须等于4，若小于4，则出错；若大于4，则只取前4个字符。CVS函数与MKS $函数的转换过程正好相反。

CVI（区段名）函数把指定区段里的2个字节的字符串转换成整型数值。指定区段里的字符串的长度必须等于2，若长度小于2就出错；若大于2，则只取前2个字符。CVI函数与MKI$ 函数的转换过程正好相反。

有关随机文件的程序例

例十一，把数据存入随机存取文件

下面的程序在随机文件DATA-2.TXT中存入10个数据（单精度数）：

```
10 OPEN "R",#1,"B:DATA-2.TXT"
20 FIELD #1,4 AS A$,4 AS B$
30 FOR I=1 TO 5
40 READ X,Y
50 LSET A$=MKS$(X)
```

```
60 LSET B$=MKS$(Y)
70 PUT #1
80 NEXT
90 CLOSE
200 DATA 3426.84,-8324.94,33135.2,-78,44.8
210 DATA -9.86,2564.21,-0.87,32.19,-99.86
```

第10行打开随机存取文件DATA-2.TXT；第20行FIELD语句把1号缓冲区划分为区段，开头4个字节命名A$，随后4个字节命名B$，当然这两个长度之和不能超过进入MBASIC时预先指定的记录长度；第50行和60行先通过MKS$函数把数据转换为4个字节的字符串，分别放到缓冲区的区段中去；第70行把1号缓冲器里内容存入当前的1号记录。

通过30—80行的循环把10个数据分别存入5个记录。

例十二，从随机文件中取出数据

从刚才建立的随机文件中取出数据的程序及运行输出结果是：

```
10 OPEN "R",#1,"B:DATA-2.TXT"
20 FIELD #1,4 AS A$,4 AS B$
30 FOR I=1 TO 5
40 GET #1,I
50 X=CVS(A$)
60 Y=CVS(B$)
70 LPRINT X,Y
80 NEXT
90 CLOSE
```

```
3426.84         -8324.94
33135.2         -78
44.8            -9.86
2564.21         -.87
32.19           -99.86
```

第40行从DATA-2.TXT中读出记录放入1号缓冲区，每循环一次读一个记录；第50行和60行分别把区段中字符串转换为单精度数值。

例十三，LSET/RSET用法

```
10 OPEN "R",#1,"B:DATA-3.TXT"
20 FIELD #1,13 AS A$,5 AS B$
30 LSET A$="123.45678"
40 LSET B$="ABCDEF"
50 LPRINT A$;B$
60 PUT #1,5
70 RSET A$="-25.33"
80 RSET B$="987.5634"
90 LPRINT A$;B$
100 PUT #1,6
110 CLOSE
123.45678    ABCDE
       -25.33987.5
```

第30行和40行用LSET语句把字符串放入缓冲区，从运行输出的A$可知在这个字符串后面留了4个空格，B$已把字符串中的F截掉；第60行把字符串存入5号记录；第70行和80行用RSET语句把字符串放入缓冲区，运行输出的A$表示在前面留了空格；第100行把字串存入6号记录。

例十四，把一批数据存入随机存取文件

若要求建立一个随机存取文件,用4个记录， 每个记录存10个数据。

```
10 DIM A$(9)
20 OPEN "R",#1,"B:DATA-4.TXT"
30 FOR I=0 TO 9
40 FIELD #1,I*4 AS D$,4 AS A$(I)
50 NEXT
60 FOR J=1 TO 4
70 FOR I=0 TO 9
80 READ X
```

```
90 LSET A$(I)=MKS$(X)
100 NEXT
110 PUT #1,J
120 NEXT
130 CLOSE
300 DATA 1,2,3,4,5,6,7,8,9,10
310 DATA 2,4,6,8,10,12,14,16,18,20
320 DATA 3,6,9,12,15,18,21,24,30,33
330 DATA 1,4,9,16,25,36,49,64,81,100
```

第40行是关键，这行用了哑自变量D$，以确定每一个区段在缓冲区中的起始位置。按设计要求每个记录存10个单精度数，即FIELD语句要预先把缓冲区分10个区段，第一次循环时，I＝0，D$长度是零，因此区段A$(0)从第1个字节开始存贮，第二次循环，I＝1，D$长度是4，故A$(1)从第5个字节开始，依此类推，第10次循环，D$长度是36，故A$(9)从第37个字节开始存贮，通过循环，把缓冲区的区段分配好了。I＊4不能写成4＊I，否则机器无法判断A$。

通过第60行到120行的双重循环把40个数分别存入4个记录。

例十五，从随机文件取出成批数据

下面程序从上例已建立的随机文件中取出数据。

```
10 DIM A$(9)
20 OPEN "R",#1,"B:DATA-4.TXT"
30 FOR I=0 TO 9
40 FIELD #1,I*4 AS D$,4 AS A$(I)
50 NEXT
```

```
60 FOR J=1 TO 4
70 GET #1,J
80 FOR I=0 TO 9
90 X=CVS(A$(I))
100 LPRINT X;
110 NEXT
120 LPRINT
130 NEXT
140 LPRINT LOF(1)
150 CLOSE
```

缓冲区的区段定位方法与例十四相同，第60行到130行的双重循环把各个记录内容读出并打印出来。第140行用到ＬＯＦ（缓冲区编号）函数，这个函数找出与指定缓冲区相连的文件中的最大记录号。这个程序运行输出：

```
1   2   3   4   5   6   7   8   9   10
2   4   6   8   10   12   14   16   18   20
3   6   9   12   15   18   21   24   30   33
1   4   9   16   25   36   49   64   81   100
4
```

例十六，其它语句用法：

```
10 OPEN "R",1,"B:DATA-5.TXT"
20 FIELD #1,4 AS A$,4 AS B$
30 PRINT#1,"ABCD";",";"123"
40 PUT#1
50 CLOSE
60 OPEN "R",1,"B:DATA-5.TXT"
70 FIELD #1,4 AS C$
80 GET #1
90 LINE INPUT#1,C$
100 CLOSE
110 LPRINT C$
```

第10行到50行建立一个随机文件ＤＡＴＡ-５.ＴＸＴ，

这里用PRINT#语句代替LSET语句把字符串放入缓冲区，类似还可用PRINT#USING，WRITE#语句代替；第60行到第110行从随机文件中取出字符串，这里用LINE INPUT#把字符串取出来。

这个程序运行输出：ABCD，123

另外在随机文件中也能用EOF（缓冲区编号）函数检查文件是否结束；能用LOC（缓冲区编号）函数在执行了GET或PUT语句之后找出记录号码。

3.6.8 GBASIC的命令和语句

进入GBASIC的工作方式

GBASIC是Microsoft BASIC的CP/M版本，它除了具有MBASIC的所有特色外，还有高分辨画图的功能，它只在16扇区的磁盘上才能使用，文件名是GBASIC.COM。

进入GBASIC的操作方法与进入MBASIC类似，只要在提示符A＞出现后键入GBASIC即可。它也有多种键入格式：

GBASIC文件名称/F：文件数目/M：存贮器的最高位置/S：记录的最大长度

各项的说明、用法与MBASIC类似。

GBASIC的命令和语句

① HGR

语法格式：

HGR屏幕号，颜色号

用于建立高分辨图形状态。

屏幕号码指定的显示状态是：

号码　是否清除画面　屏幕状态

0　是　280×160图形＋4行文字说明

1　是　280×192图形

2　否　280×160图形＋4行文字说明

3　否　280×192图形

若省略屏幕号，则认为是零；若省略颜色号，则认为颜色号是零。

② HCOLOR

语法格式：

HCOLOR＝颜色号码

指定高分辨图形显示的颜色，有13种：

0 黑色　1 绿色　2 紫罗兰色　3 白色　4 黑色　5 橙色　6 蓝色　7 白色　8 黑色1　9 白色1　10 黑色2　11 白色2　12 颜色倒转

上述颜色号码中：0，3，4，7用来画细线，用于黑白屏幕；8，9，10，11用来画粗点或粗线；8，9应与绿色或紫罗兰色合用；10，11与橙色或蓝色合用。

注意：HCOLOR只能用于高分辨率图形状态。

③ HPLOT

是画点或线的语句，用法与APPLESOFT语言的PLOT语句类似，可参照使用。

④ HSCRN

语法格式：

HSCRN(X, Y)

其中 $0 \leqslant X \leqslant 279$，$0 \leqslant Y \leqslant 191$

若在座标（X，Y）位置有点，则HSCRN(X，Y)＝－1，否则为零。

§ 3.7 MBASIC中的函数

3.7.1 算术函数

ABS(X)

取X的绝对值。X为数值表达式，以下同。

ATN(X)

求X的反正切函数值，X的单位是弧度，$-\frac{\pi}{2} \leqslant X \leqslant \frac{\pi}{2}$，ATN(X)的值是单精度值。

CDBL(X)

将X的值转变为双精度值。

CINT(X)

以四舍五入方式取X的整数，但不能超过－32768—32767的范围。

COS(X)

求X的余弦函数值，X单位为弧度，COS(X)为单精度值。

CSNG(X)

将X的值转变为单精度值。

EXP(X)

求e^X值。

FIX(X)

又一种取X的整数的函数，相当于SGN(X)＋INT.(ABS(X))

例如：FIX(12.41)＝12

FIX(－12.41)＝－12

INT(X)

取小于等于X的最大整数。

LOG(X)

取X的自然对数值，X＞0

RND(X)

产生0与1之间随机数。

SGN(X)

取X的符号。

SIN(X)

取X的正弦，X的单位是弧度，SIN（X）是单精度值。

SQR(X)

求X的平方根，X＞0

TAN(X)

求X的正切函数值，X的单位是弧度，TAN(X)是单精度值。

3.7.2 字符串函数

ASC(X$)

取X$的第1个字符的ASCII码值（X$为字符串表达式，以下同）

CHR$ (I)

表示与ASCII码的I值所相应的形式，I为整数。

例如：CHR$(66)为字符B。

CVI（2个字节的字符串）

CVS（4个字节的字符串）

CVD(8个字节的字符串）

以上三个函数都是将字符串转变为数值。从随机文件读数据，必须将字符串转变成数值，CVI函数将2个字节的字符串转

变为整数，CVS 函数将 4 个字节的字符串转变为单精度值，CVD 函数将 8 个字节的字符串转变为双精度值。详见随机文件节。

HEX$(X)

把十进制数X转换成以十六进制数表示。计算前已先把X四舍五入为整数。例如：

```
PRINT    HEX$ (32)
  20
 OK
```

LEFT$(X$, I)

从字符串X$左面起取I个字符。其中X$为字符串表达式，I为整数，I不能超过255。

LEN(X$)

求出X$的字符个数。

MID$(X$, I, J)

从字符串X$的第I个字符开始取J个字符，$0 \leqslant I \leqslant 255$，$0 \leqslant J \leqslant 255$。

MKI$（整数表达式）

MKS$（单精度表达式）

MKD$（双精度表达式）

以上三个函数分别将不同类型数值转变为字符串，用于随机文件。

OCT$（X）

把十进制数X转换为八进制数表示。计算前已先把X四舍五入为整数。例如：

```
PRINT  OCT$ (24)
30
```

OK

RIGHT$（X$，I）

从字符串X$的最右边起取I个字符。

SPACE$（X）

产生长度为X的空白字符串，X为四舍五入后的整数，$0 \leqslant X \leqslant 255$

STR$（X）

将X的值转换为字符串。

STRING$（I，J）

表示长度为I的字符串，每个字符都是J值的ASCII码。可用作命令或语句。例如：

```
10 A$=STRING$(11,45)
20 B$=STRING$(7,42)
30 LPRINT A$;"GOOD";A$
40 LPRINT B$;" BEGIN ";B$
```

运行输出：

```
-----------GOOD-----------
******* BEGIN *******
```

STRING$(I,X$)表示长度为I的字符串，每个字符都是X$的第一个字符。可用作命令或语句。例如：

```
10 A$=STRING$(4,"ABC")
20 LPRINT A$
```

运行输出：AAAA

VAL（X$）

将X$转变为数值，若X$的第一个字符不是+、-、&或数字，则VAL（X$）= 0

3.7.3 其它函数

EOF（缓冲区编号）

用于顺序处理文件和随机存取文件，是文件结束标志，可用来控制资料读取。

FRE（0）

得出未使用的内存字节数。

INKEY$

从终端键入一个字符的字符串，或不键入字符。键入CTRL-C程序就停止。

INPUT$（X，#Y）

从Y文件输入X个字符。

INPUT$(X)从终端输入X个字符。

例如下面这个程序：

```
10 A$=INPUT$(3)
20 LPRINT A$
```

运行到第10行，出现光标等待键盘输入，若键入RRR，则输出RRR，屏幕回到OK。

凡程序中包含这个函数，若在运行中键入CTRL-C,将停止运行。

INSTR（X$，Y$）

求出字符串Y$在字符串X$中第一次出现的位置。

INSTR(I，X$，Y$) 求出字符串Y$在字符串X$的位置，但从第I个字符开始查找它所处的位置。

例如下面的程序：

```
10 A$="ABCDABCDABCD"
20 B$="BC"
30 C$="AC"
40 LPRINT INSTR(A$,B$)
50 LPRINT INSTR(A$,C$)
60 LPRINT INSTR(3,A$,B$)
```

运行则打印输出：

2

0

6

LOC（缓冲区编号）

见顺序处理文件和随机文件的说明。

LOF（缓冲区编号）

对文件读或写的最后记录的数字。

LPOS （X）

找出打印机缓冲区内，打印机的打印头目前的位置。X是任意设定的参数。

PEEK （I）

从内存的I位置，得到字节资料（0至255的十进制整数），0≤I≤65536。

POS （I）

求出光标目前在显示屏上的位置。最左边的位置是1，I是任意一个数，例如下面的程序：

```
10 FOR I=1 TO 5
20 PRINT I;
30 NEXT
40 PRINT POS(1)
```

运行显示：

1 2 3 4 5 16

SCRN （X，Y）

式中X是0至39的整数，Y是0至47的整数。这个函数用于找出坐标为（X，Y）的点的颜色。

SPC （I）

显示或打印空格的函数，要求 0≤I≤255，与APPLESOFT 语言用法相同。

TAB(I)

指定显示或打印位置的函数，要求 0≤I≤255，与 APPLESOFT 语言用法相同。

USR(X)

用X值指定所调用的用户的汇编语言子程序。

VARPTR(变量名)

找出变量名第一个字节的地址。

VARPTR(#缓冲区号码) 找出设定在指定缓冲区号码的磁盘I/O缓冲区起始地址。

VPOS(I)

求出光标目前在屏幕的垂直位置，最高点是1， I 是任意值。

§ 3.8 MBASIC 的错误信息

这一节按错误码编号顺序列出MBASIC程序运行中出错的信息。

1. “NEXT without FOR” 表示程序中 NEXT 与 FOR不配对，有NEXT而无FOR。

2. “Syntax error”表示语法错，如括号不配对，语句写错等。

3. “RETURN without GOSUB” 表示程序中 RETURN与GOSUB 不配对。

4. “Out of data” 执行READ语句，没有足够的数据提供。

5. “Illegal function call” 调用函数错，例如LOG

函数中出现负值或零等。

6. “Over flow” 计算溢出,计算结果大于MBASIC的数值格式。

7. “Out of memory”程序太长，超出机器内存,或循环嵌套过多等。

8. “Undefined Line”在GOTO, GOSUB, IF…THEN …ELSE… 或DELETE语句中所指定的行号不存在。

9. “Subscript out of range”程序中用到的数组变量下标超出数组说明中的规定。

10. “Redimensioned array” 重复定义数组错。

11. “Division by zero” 表达式中除数为零错。要注意遇到这种错误，程序仍进行，但有关的值是不正确的。

12. “Illegal direct”直接命令用错。

13. “Type mismatch”变量类型不匹配错。

14. “Out of string space” 内存容量不敷字符串使用。

15. “String too long” 字符串超过255个字符。

16. “String formula too complex”字符串表达式太复杂错。

17. “Can't continue”程序无法继续执行，可能程序有错，或程序执行中暂停后对程序已作了修改等。

18. “Undefined user function”在DEF语句之前，调用USR函数错。

19. “NO RESUME” 在处理错误的程序中，后面没有RESUME语句。

20. “RESUME without error”程序中没有处理错

误语句，却用了RESUME语句。

21．“Unprintable error”在ERROR语句中指定的错误码事先没有定义。

22．“Missing operand”表达式中，只有运算符，没有运算量。

23．“Line buffer overflow”程序行中字符太多，超出每行允许范围。

26．“FOR without NEXT”程序中FOR与NEXT不配对，有FOR，无NEXT。

29．“WHILE without WEND”程序中WHILE与WEND不配对，有WHILE而无WEND。

30．“WEND without WHILE”程序中WEND与WHILE不配对，有WEND而无WHILE。

31．“Reset error”按了Reset键错，但有些APPLE Ⅱ上RESET键须与CTRL键共按，故一般不另发生这类错误。

32．“Graphics statement not implemented”在MBASIC中，不能用绘图语句，而只能用于GBASIC。

50．“Field overflow”表示Field语句分配的字节数，超出了已指定的随机文件记录长度。

51．“Internal error”磁盘MBASIC内部有错。

52．“Bad file number”用错磁盘文件号码。

53．“File not found”指定的文件不在磁盘中。

54．“Bad file mode”磁盘文件类型指定错。

55．“File already open”文件已经打开错。若打开一个已经打开的文件或Kill一个打开了的文件就出错。

57．“Disk I/O error”磁盘I/O错，操作系统无法恢

复。

58. “File already exists”用NAME命令改变磁盘文件名字时，由于原磁盘上已有同名文件，故不能执行。

61. “Dish full”磁盘空间已占满了。

62. “Input past end”输入文件语句中已无资料可输入时错。

63. “Bad record number”记录号码有错，在PUT或GET语句中，记录编号超过最大容许值（32767)或等于零时出错。

64. “Bad file name”表示在LOAD，SAVE,KILL或OPEN中，文件名用的不对，如文件名太长等。

66. “Direct statement in file”在装入一个ASCII格式的文件时，遇到了直接语句错，装入停止。

67. “Too many files”建立新文件时(用SAVE或OPEN),所有255个目录内容已满。

68. “Disk read only”磁盘只读，可能磁盘已“写保护”了，或换了磁盘没有按RESET键。

69. “Drive select error”指定的驱动器不存在。

70. “File read only”文件只供读出。已用STAT语句指定为只读的文件不能再写入。

§ 3.9 MBASIC 与 APPLESOFT 比较

3.9.1 MBASIC独有的功能

(1) MBASIC独有的命令或语句

COMMON, CALL, PRINT USING, WHILE…WEND, EDIT, AUTO, RENUM, IF…THEN…ELSE…, HSCRN(X, Y),VPOS(0)等。

(2) MBASIC 独有的函数

INSTR, HEX$, OCT$, STRING$ 等。

自定义函数中可以有多个自变量。

(3) MBASIC 独有的运算功能

AND, OR, XOR, IMP, EQV, 整除以及 MOD 等。

(4) MBASIC 有保护文件功能

MBASIC 可用保护性的二进制格式存贮文件，可以调出运行，无法列表。

(5) MBASIC 有三种数值变量形态

MBASIC 有三种数值变量形态，即整变量、单精度变量以及有16位数的双精度变量。而 APPLESOFT 只有 9 位数的精确度；而且 MBASIC 还提供了十六进制及八进制的常数。

(6) 建立数据文件有很大不同，即MBASIC 有建立磁盘文件的语句，比 APPLESOFT 简便得多。

3.9.2 两种语言共有的命令或语句

以下命令或语句在MBASIC (GBASIC)与APPLESOFT 中都适用：

GR, COLOR, PLOT VLIN, SCRN, POP, HGR, HCOLOR, HPLOT, TEXT, HTAB, INVERSE, NORMAL, PDL (0)

3.9.3 用法不同的命令或语句

以下的命令或语句在MBASIC(GMASIC)与 APPLESOFT 中用法不同：

FOR…NEXT, INPUT, ON ERROR GOTO, RESUME, TEXT, GR, HGR, CALL 等。

3.9.4 APPLESOFT 独有的功能

FLASH, STORE, RECALL, IN#, PR#, HIMEM…LOMEM, SHLOAD, XDRAW, DRAW, SCALE, 磁带LOAD, 磁带SAVE, ROT, ESC 键的屏幕编辑功能等。

§ 3.10 将文件从APPLE DOS 转移至CP/M

3.10.1 APDOS

在CP/M系统盘上有一个文件APDOS.COM，可作为过渡命令将APPLE DOS的磁盘文件转移到CP/M盘。对于数据文件（即在DOS盘上标有T的文件）与程序文件（DOS盘上标有A或I的文件）转移的步骤不同。另外若用DOS 3.2版本编成文件，在转移前必须先做扇区的转换操作。

3.10.2 数据文件转移

步骤：

(1) 把CP/M系统主盘插入A:驱动器，开机进入CP/M工作方式，显示A>。

(2) 键入APDOS 回车，显示：

APPLE Ⅱ CP/M

APPLE DOS TO CP/M FILE TRANSFER PROGRAM (C) 1980 MICROSOFT

显示提示符*（有的机器也许是:）。

(3) 在提示符后面键入命令CAT d:则把指定驱动器d:中APPLE DOS盘上的文件目录显示出来。

(4) 键入d1:文件名1 =d2:文件名2

这时显示：（设d1：是A:，d2:是B:）

INSERT APPLE DOS DISK IN DRIVE B:

INSERT CP/M DISK IN DRIVE A:

HIT RETURN TO BEGIN

即要求把DOS盘插入B:，CP/M盘插入A:，按回车键，这时显示：

WORKING…

两个驱动器交替亮灯，做文件转移工作。最后显示：

TRANSFER COMPLETE

即文件转移工作已经完成，已经把DOS盘上指定的文件（文件名2）转移到CP/M盘中的文件上了（文件名1）。

若是接着还要转移文件，驱动器不必重新指明，按上次指定的驱动器执行；若要改变驱动器，必须重新执行APDOS命令。

3.10.3 程序文件转移

步骤：

(1) 在DOS系统下，先把APPLESOFT或INTEGER BASIC程序按MBASIC编制程序语言规定改写成MBASIC程序形式。

(2) 增加程序行0行

```
0 PRINT CHR$(4);"OPEN APPLEPROG":
PRINT CHR$(4); "WRITE APPLEPROG":
POKE 33, 33 : LIST:
PRINT CHR$(4); "CLOSE":END
```

(3) RUN程序，这时在DOS盘中就会有文件APPLEPROG，这是原文件的复制品。

(4) 把含有APDOS文件的CP/M盘插入A:，开机。

(5) 键入APDOS

显示：

APPLE Ⅱ CP/M

APPLE DOS TO CP/M FILE TRANSFER PROGRAM (C) 1980 MICROSOFT

提示符*

(6) 键入CAT B：可显示B：中DOS盘目录。

(7) 键入A：文件名1＝B：APPLEPROG

显示：

INSERT APPLE DOS DISK IN DRIVE B:

INSERT CP/M DISK IN DRIVE A:

HIT RETURN TO BEGIN

即要求DOS盘插入B:，CP/M盘插入A:，按回车键进行转移，完成后显示：

TRANSFER COMPLETE

(8) 键入CTRL-C则退出APDOS。

(9) 键入MBASIC，进入MBASIC工作方式，调出磁盘文件，删除0行即全部完成转移。

第四章　程　序　库

§ 4.1　数据处理程序库

在气象、水文、地质、医学等的实际工作中，经常用到多元分析方法处理数据，这里我们选择提供了一些常用程序。数学处理方法主要取材于丁士晟所著《多元分析方法及其应用》一书，程序编排次序基本上也按该书介绍内容的先后编排，所以若是把本程序库与该书配合起来看，对于程序编制思路以及应用方法会有更清楚的了解。程序是用APPLESOFT BASIC语言编制的，只要作较小的改动，当然就可改变为MBASIC语言形式。程序以子程序方式提供，每个子程序前面说明解决的是什么问题，所用数学方法的简单介绍，子程序的使用说明，后面有简单的例题示范，读者看了子程序使用说明和例题算法，就会使用这个程序解题了。实际上读者只要编制一个输入数据的主程序，投入运行就可以了。

4.1.1　数据处理和统计检验

标准化

(1) 目的

由于各个物理量和各种因子的单位、大小很不一样，要估计一批数据的哪个因子重要，就要做标准化处理。

(2) 数学方法

如有m个因子，每个因子有n_i个数据，每个观测值记为x_{ij}，$i=1,2,\cdots,m$为因子，$j=1,2,\cdots,n_i$为样本数。

标准化后的观测值x'_{ij}是：

$$x'_{ij}=\frac{x_{ij}-\overline{x}_i}{s_i}$$

式中$\overline{x}_i$为第i个因子的平均值：

$$\overline{x}_i=\frac{1}{n_i}\sum_{j=1}^{n_i}x_{ij}$$

式中s_i为第i个因子的标准差：

$$s_i=\sqrt{\frac{\sum_{j=1}^{n_i}(x_{ij}-\overline{x}_i)^2}{n_i-1}}$$

第i个因子的变异系数为（C．V．）$_i$：

$$(C.V.)_i=s_i/\overline{x}_i$$

（3）子程序使用说明

① 简单变量

M——要求标准化处理的因子个数；

NM——第一个变量NM是诸因子中最多样本的个数；以后则代表各个因子各自观测样本的个数；

A——各个因子的平均值；

S——各个因子的标准差；

C.V.——各个因子的变异系数。

② 数组

X(NM)——输入时为各个因子观测样本的值；输出时为各个因子标准化后的值。

③ 数据排列顺序

M，NM，NM，X(NM)，NM，X(NM)，…

④ 计算结果

MEAN X——各个因子的平均值；

S——各个因子的标准差；

C.V.——各个因子的变异系数；

X(NM)——各个因子标准化后的各值。

⑤ 在主程序中按数据排列顺序给出数据后，由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM STANDARD(AP-1)
1002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
1004  READ M,NM:NM = NM - 1
1006  DIM X(NM)
1008  FOR I = 1 TO M
1010  READ NM:NM = NM - 1
1012 A = 0:B = 0
1014  FOR J = 0 TO NM
1016  READ X(J)
1018 A = A + X(J)
1020 B = B + X(J) ^ 2
1022  NEXT J
1024 S =  SQR ((B - A ^ 2 / (NM +
     1)) / NM)
1026 A = A / (NM + 1)
1028 CV = S / A
1030  PRINT "MEAN X = "; FN P(A);
      TAB( 20);"S = "; FN P(S); TAB(
     40);"C.V. = "; FN P(CV)
1032  FOR J = 0 TO NM STEP 10
1034 C = NM - J
1036  IF C > 9 THEN C = 9
1038  FOR K = J TO J + C
1040 X(K) = (X(K) - A) / S
1042  PRINT  FN P(X(K)); SPC( 2);

1044  NEXT K
1046  PRINT
```

```
1048  NEXT J
1050  PRINT
1052  NEXT I
1054  RETURN
```

（5）例题

① 如有两个序列x_1和x_2，样本长度n＝9，即是：

$x_1=(5,6,8,7,7,4,6,5,6)'$

$x_2=(0.8,0.3,0.6,0.7,0.5,0.4,0.7,0.2,0.3)'$

求两个序列标准化处理后的值。

② 主程序

```
10   DATA 2,9
20   DATA 9,5,6,8,7,7,4,6,5,6
30   DATA 9,0.8,0.3,0.6,0.7,0.5,0.
      4,0.7,0.2,0.3
40   GOSUB 1000
50   END
```

③ 标准化子程序（AP-1）

④ 计算结果：

```
ÜRUN
MEAN X = 6           S = 1.22               C.V. = .2
-.82  0  1.63  .82  .82  -1.63  0  -.82  0

MEAN X = .5          S = .21                C.V. = .42
1.41  -.94  .47  .94  0  -.47  .94  -1.41  -.94
```

t检验（检验样本与母体平均数的差异）

（1）目的

检验一个小样本的平均数是否来自已知平均数的正态母体。

（2）数学方法

已知正态母体的子样为x_1，x_2，…，x_n，母体的平均数是m_0。

则子样平均数是：

$$\bar{x}=\sum_{i=1}^{n} x_i/n$$

子样标准差是：

$$S=\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2/(n-1)}$$

检验第二类错误所需的统计量：

$$DT=(\bar{x}-m_0)/S$$

统计量 t：

$$t=DT\sqrt{n}$$

给定信度 α，从数理统计表上查 t 分布表，得到置信度 t_α，若 $|t|\geqslant t_\alpha$，说明子样平均值与 m_0 差异显著；若 $|t|<t_\alpha$，则没有显著差异。给定信度 α，样本数，DT值，查第二类错误的概率。

(3) 子程序使用说明

① 简单变量

N——样本个数；

M——母体的总体数学期望值；

X——样本数据；

DT，T——统计量。

② 数据排列顺序

N，M，X

③ 计算结果

打印输出统计量T 和DT 。

④ 在主程序中按数据排列顺序给出数据后，GOSUB

1000转本节子程序。

(4) 子程序

```
1000  REM T-TEST(AP-2)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ N,M
1006  FOR I = 1 TO N
1008  READ X
1010 SA = SA + X
1012 SP = SP + X ^ 2
1014  NEXT I
1016 DT = (SA / N - M) /  SQR ((S
     P - SA ^ 2 / N) / (N - 1))
1018 T = DT *  SQR (N)
1020  PRINT "T="; FN P(T)
1022  PRINT "DT="; FN P(DT)
1024  RETURN
```

(5) 例题

① 如有两种观测方法，同样测量10次，这两种方法测量误差的差值如下（A－B）所示：

(A－B)=(1, 3, 1, 1, 0, 1, 2, 1,5,1)

问：这两种观测方法之间有没有差别？

② 可设两种观测方法之间没有不同，即设$m_0=0$，则可有如下主程序：

```
10   DATA 10,0
20   DATA 1,3,1,1,0,1,2,1,5,1
30   GOSUB 1000
40   END
```

③ t检验子程序（AP-2）

④ 计算结果：

```
URUN
T=3.539
DT=1.119
```

给定信度 $\alpha=0.05$，查 t 分布表得 $t_{0.05}=2.26$（自由度 $f=n-1=9$）；若给定信度 $\alpha=0.01$，则 $f=9$ 时 $t_{0.01}=3.25$。

由于 $T>t_{0.05}$，$T>t_{0.01}$，说明子样平均值与 m_0 差异显著，即两种观测方法之间差异是明显的。

已知：$DT=1.11$，$n=10$，$\alpha=0.05$，可查得 $\beta=0.20$；

已知：$DT=1.11$，$n=10$，$\alpha=0.01$，可查得 $\beta=0.50$；

所以综合考虑应该取具有第一类错误 $\alpha=0.05$；具有第二类错误 $\beta=0.20$ 为好。

t 检验（检验两个样本平均数的差异）

（1）目的

检验两个小样本的平均数差异是否显著。

（2）数学方法

已知两个正态变量母体的子样为：$x_1, x_2, \cdots, x_{n1}$ 和 $y_1, y_2, \cdots, y_{n2}$

两个子样的平均数分别是：

$$\overline{x}=\sum_{i=1}^{n} x_i/n_1$$

$$\overline{y}=\sum_{i=1}^{n} y_i/n_2$$

两个子样的标准差分别是：

$$s_x=\sqrt{\frac{\sum_{i=1}^{n_1}(x_i-\overline{x})^2}{n_1-1}}$$

$$s_y=\sqrt{\frac{\sum_{i=1}^{n_2}(y_i-\overline{y})^2}{n_2-1}}$$

统计量DT是：

$$DT=\frac{\overline{x}-\overline{y}}{\sqrt{\frac{(n_1-1)s_x^2+(n_2-1)s_y^2}{n_1+n_2-2}}}$$

统计量T是：

$$T=\frac{DT}{\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$$

自由度F是：

$$F=n_1+n_2-2$$

给定信度 α，由F值从数理统计表可查置信限 t_α，若 $|T|\geqslant t_\alpha$，表明 $\overline{x}$ 与 $\overline{y}$ 差异显著；$|T|<t_\alpha$，则差异不显著。由DT值，给定 α，样本个数，可查出第二类错误的 β 值。

(3) 子程序使用说明

① 简单变量

N1——第一个子样的样本个数；

N 2——第二个子样的样本个数；

X——前面代表第一个子样的各个样本数值，后面代表第二个子样的各个样本数值；

F——自由度；

DT，T——统计量。

② 数据排列顺序

N1，N2，X

③ 计算结果

打印输出自由度及统计量。

④ 在主程序中按数据排列顺序给出数据后，由GOSUB 1000转本节子程序。

（4）子程序

```
1000  REM T-TEST(AP-3)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ N1,N2
1008  FOR I = 1 TO N1
1010  READ X
1012 S1 = S1 + X:S2 = S2 + X ^ 2
1014  NEXT I
1016  FOR I = 1 TO N2
1018  READ X
1020 S3 = S3 + X:S4 = S4 + X ^ 2
1022  NEXT I
1024 SX =  SQR ((S2 - S1 ^ 2 / N1
     ) / (N1 - 1))
1026 SY =  SQR ((S4 - S3 ^ 2 / N2
     ) / (N2 - 1))
1028 DT = (S1 / N1 - S3 / N2) /  SQR
     (((N1 - 1) * SX ^ 2 + (N2 -
     1) * SY ^ 2) / (N1 + N2 - 2)
     )
1030 T = DT /  SQR (1 / N1 + 1 /
```

```
        N2)
1032  PRINT "F=";N1 + N2 - 2
1034  PRINT "DT="; FN P(DT)
1036  PRINT "T="; FN P(T)
1038  RETURN
```

（5）例题

① 如某年搞10次人工降雨，得到作业区（X），对照区（Y）雨量，问人工降雨效果是否显著？

X＝（7，3，2，4，6，6，7，5，6，4）

Y＝（3，6，2，2，1，4，2，6，3，1）

② 主程序

```
10  DATA 10,10
20  DATA 7,3,2,4,6,6,7,5,6,4
30  DATA 3,6,2,2,1,4,2,6,3,1
40  GOSUB 1000
50  END
```

③ t检验子程序（AP-3）

④ 计算结果：

```
]RUN
F=18
DT=1.134
T=2.535
```

取信度 $\alpha=0.05$，F＝18，则 $t_{0.05}=2.10$，由于 $T>t_{0.05}$，说明 $\overline{x}$ 与 $\overline{y}$ 差异显著，人工降雨效果明显。DT＝1.13，N＝10，$\alpha=0.05$时 $\beta=0.20$。

χ^2检验（检验样本标准差与母体均方差的差异）

（1）目的

检验样本的标准差与它母体均方差之间有无差异。

(2) 数学方法

已知母体数学期望$\bar{x}$，均方差σ，它的子样是$x_1, x_2, \cdots, x_n$

样本离差平方和：

$$SD = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

统计量χ^2是：

$$\chi^2 = SD/\sigma$$

子样标准差平方是：

$$S^2 = SD/(n-1)$$

统计量R是：

$$R = S^2/\sigma^2$$

自由度F是：

$$F = n-1$$

已知F，给定信度α，由数理统计表可查χ_α^2表，得置信限χ_α^2，若算得的$\chi^2 \geqslant \chi_\alpha^2$，则说明子样与母体的分布有明显差异；若是$\chi^2 < \chi_\alpha^2$，说明子样与母体的分布没有明显差异。由R，$\alpha$和F值，可以查得出第二类错误$\beta$值。

(3) 子程序使用说明

① 简单变量

XP——母体的数学期望；

DP——母体的均方差；

N——子样的个数；

X——子样的各个数值；

KA，R——统计量。

② 数据排列顺序

XP, DP, N, X

③ 计算结果

打印输出:

KAPA∧2——统计量χ^2;

R——统计量R;

F——自由度。

④ 在主程序中按数据排列顺序给出数据后，由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM KAPA-TEST(AP-4)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ XP,DP,N
1006  FOR I = 1 TO N
1008  READ X
1010 S1 = S1 + X
1012 S2 = S2 + X ^ 2
1014  NEXT I
1016 SD = S2 - 2 * S1 * XP + N *
     XP ^ 2
1018 KA = SD / DP
1020 R = SD / ((N - 1) * DP ^ 2)
1022  PRINT "KAPA^2="; FN P(KA)
1024  PRINT "R="; FN P(R)
1026  PRINT "F=";N - 1
1028  RETURN
```

(5) 例题

① 如某水库多年平均月库容为4亿方，标准差为2亿方，最近5年的月平均库容分别为4，3，0，6，7亿方，问这5年库容变动有没有反常?

② 主程序

```
10   DATA 4,2,5
20   DATA 4,3,0,6,7
30   GOSUB 1000
40   END
```

③ χ^2检验子程序(AP-4)

④ 计算结果：

```
]RUN
KAPA^2=15
R=1.875
F=4
```

由F＝4，信度α＝0.05时，从数理统计表查得$\chi^2_{0.05}$＝9.488；

由F＝4，α＝0.05，R＝1.87时，查得β＝0.5从以上检验说明，子样分布可能与多年分布不一样，出第一类错误的概率为5％，出第二类错误的概率为50％，所以只根据这些有限的资料还不足以判断这5年的库容反常。

χ^2检验（检验联列表中因子相互独立性）

(1) 目的

检验联列表中因子相互独立性。

(2) 数学方法

设有两个因子x和y，x分为r类，y分为S类，n_{ij}代表两个因子各种组合观测次数。

$$S1=\sum_{i=1}^{s} n_{ij}$$

$$S2=\sum_{i=1}^{r} n_{ij}$$

$$SS=\sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij}$$

理论次数为:

$$nP_{ij}=S1\cdot S2/SS$$

统计量χ^2是:

$$\chi^2=\sum_{r\times s}\frac{(n_{ij}-nP_{ij})^2}{nP_{ij}}$$

若观测次数少，须加连续性订正，即:

$$\chi^2=\sum_{r\times s}\frac{(|n_{ij}-nP_{ij}|-0.5)^2}{nP_{ij}}$$

在本程序设计中，假设观测次数已足够多，故计算 χ^2 的表达式中没有加连续性订正项。自由度f 为:

$$f=(r-1)\times(s-1)$$

已知自由度 f ，指定信度α，可由如下近似公式计算χ^2值，以省去查找数理统计表的麻烦。当$\alpha=0.01$时:

$$\chi^2_{0.01}=f+2.2+2.326\sqrt{2f-1}$$

当$\alpha=0.05$时:

$$\chi^2_{0.05}=f+0.85+1.645\sqrt{2f-1}$$

若$\chi^2\geqslant\chi^2_\alpha$，则x与y不独立，有一定关系，否则x与y独立，没有密切关系。

(3) 子程序使用说明

① 简单变量

N1——因子 x 分成的类数；

N2——因子 y 分成的类数；

F——自由度；

KA，K1，K2——统计量。

② 数组

X(N1，N2)——联列表中各组合观测值；

S1(N1)——因子x各类观测总数；

S2(N2)——因子y各类观测总数。

③ 数据排列顺序

N1，N2，X(N1，N2)

④ 计算结果

打印输出：

F——自由度；

KAPA∧2(0.01)——信度0.01时的χ^2值；

KAPA∧2(0.05)——信度0.05时的χ^2值；

KAPA∧2——计算的χ^2值；

若信度为0.01时，x 与 y 关系显著，则打印“alfa = 0.01 pass”；

若信度为0.05时，x 与 y 关系显著，则打印“alfa=0.05 pass”；

若信度为0.05时，x与y关系不显著，则打印“alfa = 0.05 no pass”。

⑤ 在主程序中，按数据排列顺序给出数据后，由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM KAPA-TEST(AP-5)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ N1,N2:N1 = N1 - 1:N2 =
     N2 - 1
1006  DIM X(N1,N2),S1(N1),S2(N2)
1008  FOR I = 0 TO N1
1010  FOR J = 0 TO N2
1012  READ X(I,J)
1014 S1(I) = S1(I) + X(I,J)
1016 S2(J) = S2(J) + X(I,J)
1018  NEXT J
1020 SS = SS + S1(I)
1022  NEXT I
1024  FOR I = 0 TO N1
1026  FOR J = 0 TO N2
1028 KA = KA + (X(I,J) - S1(I) *
     S2(J) / SS) ^ 2 * SS / (S1(I
     ) * S2(J))
1030  NEXT J
1032  NEXT I
1034 F = N1 * N2
1036 K1 = F + 2.2 + 2.326 *  SQR
     (2 * F - 1)
1038 K2 = F + 0.85 + 1.645 *  SQR
     (2 * F - 1)
1040  PRINT "F=";F
1042  PRINT "KAPA^2(0.01)="; FN P
     (K1); TAB( 25);"KAPA^2(0.05)
     ="; FN P(K2)
1044  PRINT "KAPA^2="; FN P(KA)
1046  IF KA > K1 THEN  PRINT "alf
     a=0.01    pass": GOTO 1052
1048  IF KA > K2 THEN  PRINT "alf
     a=0.05    pass": GOTO 1052
1050  PRINT "alfa=0.05    no pass
     "
1052  RETURN
```

(5) 例题

① 某气象台把一些预报指标综合为预报指标权数，预报降水情况如下表：

	无雨	小雨	中雨	大雨
指标⩾5	20	5	5	0
指标=±4	5	20	10	5
指标⩽−5	5	5	10	10

这是3×4联列表。问预报指标与降雨有没有关系？

② 主程序

```
10   DATA 3,4
20   DATA 20,5,5,0
30   DATA 5,20,10,5
40   DATA 5,5,10,10
50   GOSUB 1000
60   END
```

③ χ^2检验子程序

④ 计算结果：

```
URUN
F=6
KAPA^2(0.01)=15.914          KAPA^2(0.05)=12.306
KAPA^2=41.25
alfa=0.01        pass
```

由于$\chi^2>\chi^2_{0.01}$，说明预报指标与降雨情况有关，即指标有一定预报能力。

F检验（检验两个母体方差的差异）

(1) 目的

检验两个正态母体的方差是否有显著的差异。

(2) 数学方法

设有两个正态母体$N(\mu_1, \sigma_1^2)$和$N(\mu_2, \sigma_2^2)$，要求比较两个母体子样的标准差S1和S2，样本分别为x_1，x_2，…，x_{n1}和Y_1，Y_2，…，Y_{n2}

两个样本的标准差分别是：

$$S_1^2=\frac{1}{n_1-1}\sum_{i=1}^{n1}(x_i-\bar{x})^2$$

$$S_2^2=\frac{1}{n_2-1}\sum_{i=1}^{n2}(Y_i-\bar{Y})^2$$

统计量F 为：

$$F=S_1^2/S_2^2\simeq\lambda=\sigma_1^2/\sigma_2^2$$

自由度分别为：

$$f_1=n_1-1$$

$$f_2=n_2-1$$

由f_1和f_2以及给定信度α，可以查得置信限F_α；由f_1，f_2，α，$F(\simeq\lambda)$值，可查β值。若$F\geqslant F_\alpha$，则两母体子样标准差有明显差异；若$F<F_\alpha$，则没有明显差异。

(3) 子程序使用说明

① 简单变量

N1——x样本的个数；

N2——y样本的个数；

X——x样本的各个数值；

Y——y样本的各个数值；

F——统计量

② 数据排列顺序

N1，N2，X，Y

③ 计算结果

打印输出第一自由度(N1－1)和第二自由度(N2－1)下的统计量F值。

④ 在主程序中按数据排列顺序给出数据后，由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM F-TEST(AP-6)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ N1,N2
1006  FOR I = 1 TO N1
1008  READ X
1010 S1 = S1 + X
1012 S2 = S2 + X ^ 2
1014  NEXT I
1016  FOR I = 1 TO N2
1018  READ Y
1020 S3 = S3 + Y
1022 S4 = S4 + Y ^ 2
1024  NEXT I
1026 SA = (S2 - S1 ^ 2 / N1) / (N
     1 - 1)
1028 SB = (S4 - S3 ^ 2 / N2) / (N
     2 - 1)
1030  IF SA < SB THEN F = SB / SA
     : GOTO 1034
1032 F = SA / SB
1034  PRINT "F(";N1 - 1;",";N2 -
     1;")="; FN P(F)
1036  RETURN
```

(5) 例题

① 已知某地1月的平均气温为－12℃，－14℃，－17℃，－15℃，－13℃，－9℃，－19℃，－17℃，－17℃，－14℃，

－16℃；7月的平均气温为21℃，22℃，22℃，21℃,23℃，24℃，23℃，21℃，21℃，21℃，21℃，问这两个月气温的方差有无明显差异？

② 主程序

```
10   DATA 11,11
20   DATA -12,-14,-17,-15,-13,-9,-
      19,-17,-17,-14,-16
30   DATA 21,22,22,21,23,24,23,21,
      21,21,21
40   GOSUB 1000
50   END
```

③ F检验子程序（AP-6）

④ 计算结果

```
ÜRUN
F(10,10)=6.844
```

由$f_1=10$，$f_2=10$，$\alpha=0.05$时$F_{0.05}=2.98$；由$f_1=10$，$f_2=10$，$\alpha=0.05$，$\beta=0.10$时$\lambda=6.917$，现在算得$F=6.84$，所以β近似为0.12，即1月气温和7月气温的方差有明显差异，出第一类错误的概率为5％，出第二类错误的概率为12％。

F检验（检验多个母体数学期望值的差异）

(1) 目的

检验多个正态母体的数学期望是否有明显差异。

(2) 数学方法

如有样本x，分为k类，假设它们的均值相等。

分类平均值为：

$$\bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$$

总平均值为：

$$\bar{x} = \frac{1}{k} \sum_{i=1}^{k} \bar{x}_i$$

类内平均平方和为：

$$S^2 = \frac{1}{k} \sum_{i=1}^{k} \left[\frac{1}{(n_i - 1)} \cdot \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 \right]$$

类间平均平方和为：

$$S'^2 = \frac{1}{k-1} \sum_{i=1}^{k} n_i (\bar{x}_i - \bar{x})^2$$

方差比为：

$$F = S'^2 / S^2$$

自由度为：

$$f_1 = k - 1$$
$$f_2 = n - k$$

其中n为总次数。

由f_1，f_2，给定α值，从数理统计表查得置信限F_α，若$F \geq F_\alpha$，则表示各类平均值有明显差异；若$F < F_\alpha$，则各类平均值没有明显差异。

为查得出第二类错误的概率β值，须求非中心参数值ψ：

$$\psi \simeq \sqrt{F}$$

由f_1，f_2，α和ψ值，从数理统计表查β值。

(3) 子程序使用说明

① 简单变量

M——样本分成的组数；

X——样本数值；

SP——总的样本个数；

F，PC——统计量。

② 数组

N（M）——每组样本的样本个数；

SI（M）——每组样本数值和。

③ 数据排列顺序

M，N（M），X，…

④ 计算结果

打印输出：

F（f_1，f_2）——第一自由度f_1，第二自由度f_2之下的统计量F值；

PC——统计量。

⑤ 在主程序中按数据排列顺序给出数据后，由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM F-TEST(AP-7)
1002  DEF  FN P(Y) =  INT (Y * 10
      00 + 0.5) / 1000
1004  READ M:M = M - 1
1006  DIM N(M),S1(M)
1008  FOR I = 0 TO M
1010  READ N(I)
```

```
1012 S2 = 0
1014  FOR J = 1 TO N(I)
1016  READ X
1018 S1(I) = S1(I) + X
1020 S2 = S2 + X ^ 2
1022  NEXT J
1024 SA = SA + (S2 - S1(I) ^ 2 /
     N(I)) / (N(I) - 1)
1026 S1(I) = S1(I) / N(I)
1028 SS = SS + S1(I)
1030  NEXT I
1032 SS = SS / (M + 1)
1034 SA = SA / (M + 1)
1036  FOR I = 0 TO M
1038 SB = SB + N(I) * (S1(I) - SS
     ) ^ 2
1040 SP = SP + N(I)
1042  NEXT I
1044 SB = SB / M
1046 F = SB / SA
1048 PC =  SQR (F)
1050  PRINT "F(";M;",";SP - M - 1
     ;")="; FN P(F)
1052  PRINT "PC="; FN P(PC)
1054  RETURN
```

(5) 例题

① 某水文站流量1951—1970年的距平值如下：5，－5，0，3，－10，－3，9，－4，2，5，－1，－1，－5，8，5，8，－8，－2，3，－8，问三年周期是否明显？

② 主程序

因为计算的是3年周期，故M＝3，并把数据按三年周期先排列好，以便写入主程序，数据排列成：

5，－5，0

3，－10，－3

9， －4， 2

5， －1， －1

－5， 8， 5

8， －8， －2

3， －8

下面就是主程序：

```
10   DATA 3
20   DATA 6,0,-3,2,-1,5,-2
30   DATA 7,-5,-10,-4,-1,8,-8,-8
40   DATA 7,5,3,9,5,-5,8,3
50   GOSUB 1000
60   END
```

③ F检验子程序（AP-7）

④ 计算结果

```
URUN
F(2,17)=5.05
PC=2.247
```

由$f_1=2$，$f_2=17$，$\alpha=0.05$时$F_{0.05}=3.59$； $\alpha=0.01$时$F_{0.01}=6.11$。

因为$F=5.05>F_{0.05}$， 表示可能存在3年周期， 信度是5％； $\psi=2.24$时，查得$\beta\simeq0.25$，即可能存在3年周期的判断，出第一类错误的概率为5％，出第二类错误的概率为25％。

4.1.2 代数运算及特殊函数

阶乘

(1) 目的

计算正整数的阶乘。

(2) 数学方法

数值N的阶乘是：

$$N! = N \times (N-1) \times (N-2) \times \cdots \times 3 \times 2 \times 1$$

(3) 子程序使用说明

① 简单变量

N——要计算阶乘的数据；

K——计算出阶乘的值。

② 数据排列顺序

N

③ 计算结果

打印输出要求的阶乘值。

④ 在主程序的数据语句中，给出N的具体数值后，由GO-SUB 1000转本节子程序。

(4) 子程序

```
1000  REM N!(AP-8)
1002  READ N
1004 K = 1
1006  FOR I = 1 TO N
1008 K = K * I
1010  NEXT I
1012  IF N = 0 THEN  PRINT "0!=1"
     : END
1014  PRINT N;"!=";K
1016  RETURN
```

(5) 例题

① 求10的阶乘。

② 主程序

```
10  DATA 10
20  GOSUB 1000
30  END
```

③ 求阶乘子程序（AP-8）

④ 计算结果

10！=3628800

Γ函数

(1) 目的

计算实数的Γ函数。

(2) 数学方法

① 计算Γ函数

$$\Gamma(x)=\int_0^{\infty} e^{-t} t^{x-1} dt$$

其范围 $0\leqslant x\leqslant 3$，利用切比雪夫逼近法：

$$\Gamma(2+x)=\sum_{k=0}^{n} a_k x^k+\varepsilon_n(x)$$

$$\varepsilon_{max}=\max_{0\leqslant x\leqslant 1}|\varepsilon_n(x)|$$

取n=10时，则 $\varepsilon_{max}\leqslant 10^{-10}$，精确度一般达到要求。

其中：

$a_0=1.0$

$a_1=0.42278433$

$a_2=0.41184025$

$a_3=0.081578218$

$a_4=0.074237907$

$a_5=-0.0002109075$

$a_6=0.010973695$

$a_7=-0.0024667480$

$a_8=0.0015397681$

$a_9 = -0.0003442342$

$a_{10} = 0.0000677106$

② 利用递推公式Γ（x+1）=xΓ(x) 求解。

(3) 子程序使用说明

① 简单变量

Z ——给定的自变量；

G ——Z 的Γ函数。

② 数据排列顺序

Z

③ 计算结果

打印输出Z 的Γ函数值。

④ 在主程序的数据语句中,给出具体Z 值，由GOSUB 9000转本节子程序。

(4) 子程序

```
9000  REM GAMA(X)(AP--9)
9002  DEF  FN P(Y) =  INT (Y * 10
    0 + 0.5) / 100
9004  DIM A(10)
9006 A(0) = 1
9008 A(1) = 0.42278433
9010 A(2) = 0.41184025
9012 A(3) = 0.081578217
9014 A(4) = 0.074237907
9016 A(5) =  - 0.0002109075
9018 A(6) = 0.010973695
9020 A(7) =  - 0.0024667480
9022 A(8) = 0.0015397681
9024 A(9) =  - 0.0003442342
9026 A(10) = 0.0000677106
9028  READ Z
9030  IF Z <  = 1 THEN 9042
9032  IF Z <  = 2 THEN 9038
```

```
9034 T = Z - 2
9036  GOTO 9044
9038 T = Z - 1
9040  GOTO 9044
9042 T = Z
9044 P = A(10)
9046  FOR K = 9 TO 0 STEP  - 1
9048 P = T * P + A(K)
9050  NEXT K
9052  IF Z <  = 1 THEN 9064
9054  IF Z <  = 2 THEN 9060
9056 G = P
9058  GOTO 9066
9060 G = P / Z
9062  GOTO 9066
9064 G = P / (Z * (Z + 1))
9066  RETURN
```

(5) 例题

① 求Z＝2.4时的Γ函数值。

② 主程序

```
10  DATA 2.4
20  GOSUB 9000
30  PRINT "GM(";Z;")="; FN P(G
40  END
```

③ 求Γ函数的子程序（AP－9）

④ 计算结果

```
URUN
GM(2.4)=1.24
```

Γ函数的对数

(1) 目的

x＞0 时，计算Γ（x）的自然对数。

(2) 数学方法

利用司梯林（Stirling）公式：

$$\ln\Gamma(x)=(x-\frac{1}{2})\ln x-x+\frac{1}{2}\ln(2\pi)+\frac{1}{12x}$$

$$-\frac{1}{360x^3}+\frac{1}{1260x^5}-\frac{1}{1680x^7}+\cdots$$

(3) 子程序使用说明

① 简单变量

X——给定的自变量；

L——存放计算结果。

② 数据排列顺序

X

③ 计算结果

打印输出x的Γ函数的对数值。

④ 在主程序中给出数据后，由GOSUB 9000转本节子程序。

(4) 子程序

```
9000  REM LGAMA(X)(AP-10)
9002  DEF  FN P(Y) =  INT (Y * 10
      0 + 0.5) / 100
9004 X = X0
9006  IF X > = 7 THEN 9028
9008 F = 1.0
9010 Z = X
9012  IF Z > = 7 THEN 9022
9014 X = Z
9016 F = F * Z
9018 Z = Z + 1
9020  GOTO 9012
9022 X = X + 1
9024 F =  -  LOG (F)
9026  GOTO 9030
9028 F = 0
```

```
9030 Z = 1 / X ^ 2
9032 A = (( - 0.00059523809 * Z +
     0.00079365079) * Z - 0.00277
     77778) * Z
9034 L = F + (X - 0.5) *  LOG (X)
      - X + 0.91893853 + (A + 0.0
     83333333) / X
9036  RETURN
```

(5) 例题

① 求当x=1.5,3.0,7.5,10时的Γ函数的对数。

② 主程序

```
10   DATA 1.5,3.0,7.5,10
20   FOR K = 1 TO 4
30   READ X0
40   GOSUB 9000
50   PRINT "LGM(";X0;")="; FN P(L)

60   NEXT K
70   END
```

③ Γ函数对数的子程序（AP-10）

④ 计算结果

```
]RUN
LGM(1.5)=-.12
LGM(3)=.69
LGM(7.5)=7.53
LGM(10)=12.8
```

LGM (x) 即为x的Γ函数的对数值。

4.1.3 线性代数计算

求矩阵元素的极大值与极小值

(1) 目的

对于N行M列矩阵

$$A=\begin{bmatrix} a_{11}, & a_{12}, & \cdots, & a_{1m} \\ a_{21}, & a_{22}, & \cdots, & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1}, & a_{n12}, & \cdots, & a_{nm} \end{bmatrix}$$

求其元素的极大值与极小值。

(2) 数学方法

$$B=\max(a_{ij})$$

$$C=\min(a_{ij})$$

式中$i=1,2,\cdots,n$;

$j=1,2,\cdots,m$。

(3) 子程序使用说明

① 简单变量

N——给定矩阵A的行数;

M——给定矩阵A的列数;

B——存放矩阵元素的极大值;

C——存放矩阵元素的极小值。

② 数组

A(N-1,M-1)——存放矩阵的全部元素。

③ 数据排列顺序

N, M, A(N-1, M-1)。

④ 计算结果

极大值存在B中,极小值存在C中,打印输出。

⑤ 在主程序的数据语句中,按数据排列顺序给出数据,然后由GOSUB 9000转本节子程序。

(4) 子程序

```
9000  REM SUB1(AP-11)
9002  READ N,M
```

```
9004 N = N - 1
9006 M = M - 1
9008  DIM A(N,M)
9010  FOR I = 0 TO N
9012  FOR J = 0 TO M
9014  READ A(I,J)
9016  NEXT J
9018  NEXT I
9020 C = A(0,0)
9022 B = A(0,0)
9024  FOR I = 0 TO N
9026  FOR J = 0 TO M
9028  IF A(I,J) <  = B THEN 9034
9030 B = A(I,J)
9032  GOTO 9038
9034  IF A(I,J) >  = C THEN 9038
9036 C = A(I,J)
9038  NEXT J
9040  NEXT I
9042  PRINT "max=";B
9044  PRINT "min=";C
9046  RETURN
```

(5) 例题

① 给出矩阵

$$A=\begin{bmatrix} 1 & 0 & 0 & 5 & -36 \\ 10 & 3 & 4 & 10 & -27 \\ 8 & 6 & 5 & 7 & -18 \\ 4 & 9 & -12 & 1 & 4 \\ 0 & 1 & 3 & -60 & 9 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

求元素的极大值与极小值。

② 主程序

```
10   DATA 6,5,1,0,0,5,-36,10,3,4,1
      0
20   DATA -27,8,6,5,7,-18,4,9,-12,
      1,4
30   DATA 0,1,3,-60,9,1,2,3,4,5
40   GOSUB 9000
50   END
```

③ 挑选矩阵元素极值的子程序（AP-11）

④ 计算结果

```
URUN
max=10
min=-60
```

求行列式的值

(1) 目的

计算n阶实数行列式的值。

(2) 数学方法

采用扫除法把矩阵化为三角矩阵。

设矩阵是$\{a_{ij}\}$　$(i, j = 1—n)$

$P = a_{mm}$　$(m = 2—n)$

$q = a_{im}/P$　$(i = 1—m-1)$

$a_{ij} = a_{ij} - q \cdot a_{mj}$　$(j = 1—m)$

计算得出$a_{ij} = 0$　$(i < j)$

因此，行列式的值$DET = a_{11} \cdot a_{22} \cdot a_{33} \cdots a_{nn}$

若在计算中$P = 0$则打印出错误讯息，计算中止。

(3) 子程序使用说明

① 简单变量

N——行列式的阶数；

D——存放行列式的数值。

② 数组

A（N－1，N－1）——存放行列式的全部元素。

③ 数据排列顺序

N，A（N－1，N－1）

④ 计算结果

打印输出行列式的值DET。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB 9000转本节子程序。

(4) 子程序

```
9000  REM DETERMINANT(AP-12)
9002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
9004  READ N
9006 N = N - 1
9008  DIM A(N,N)
9010  FOR I = 0 TO N
9012  FOR J = 0 TO N
9014  READ A(I,J)
9016  NEXT J
9018  NEXT I
9020  FOR M = N TO 1 STEP  - 1
9022 P = A(M,M)
9024  IF P = 0 THEN 9052
9026  FOR I = 0 TO M - 1
9028 Q = A(I,M) / P
9030  FOR J = 0 TO M
9032 A(I,J) = A(I,J) - Q * A(M,J)

9034  NEXT J
9036  NEXT I
9038  NEXT M
9040 D = A(0,0)
9042  FOR I = 1 TO N
9044 D = D * A(I,I)
9046  NEXT I
```

```
9048  PRINT "DET="; FN P(D)
9050  RETURN
9052  PRINT "ERROR": END
```

(5) 例题

① 给出行列式

$$\begin{vmatrix} 4 & 7 & 1 & 8 \\ 5 & -1 & 2 & -4 \\ 3 & 12 & -5 & 6 \\ 1 & 4 & 7 & 2 \end{vmatrix}$$

计算它的数值。

② 主程序

```
10   DATA 4,4,7,1,8,5,-1,2,-4
20   DATA 3,12,-5,6,1,4,7,2
30   GOSUB 9000
40   END
```

③ 求行列式的子程序（AP-12)

④ 计算结果

DET =－3276

求逆矩阵

(1) 目的

计算 n阶矩阵的逆矩阵

(2) 数学方法

设给定矩阵是 $A=\{a_{ij}\}$ $(i, j=1-n)$

$a_{ii}=a_{ii}+1$ $(i=1-n)$

$P=a_{mm}-1$ $(m=1-n)$

$a_{mj}=a_{mj}/P$ $(j=1-n)$

$a_{ij}=a_{ij}-a_{im}a_{mj}$ $(i=1-n, j=1-n)$

$a_{ii}=a_{ii}-1 \quad (i=1-n)$

由以上转换得出的是原矩阵的逆矩阵。若转换中 P＝0 则中止计算，打印出错误讯息。

(3) 子程序使用说明

① 简单变量

N——矩阵的阶数。

② 数组

A（N－1， N－1）——起始存放原始矩阵的元素，最后存放逆矩阵的元素。

③ 数据排列顺序

N， A（N－1， N－1）。

④ 计算结果

打印输出逆矩阵的全部元素。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOS UB 9000转本节子程序。

(4) 子程序

```
9000  REM INVERSE MATRIX(AP-13)
9002  READ N
9004 N = N - 1
9006  DIM A(N,N)
9008  FOR I = 0 TO N
9010  FOR J = 0 TO N
9012  READ A(I,J)
9014  NEXT J
9016  NEXT I
9018  FOR I = 0 TO N
9020 A(I,I) = A(I,I) + 1
9022  NEXT I
9024  FOR M = 0 TO N
9026 P = A(M,M) - 1
9028  IF P = 0 THEN 9070
9030  FOR J = 0 TO N
```

```
9032 A(M,J) = A(M,J) / P
9034  NEXT J
9036  FOR I = 0 TO N
9038  IF I = M THEN 9048
9040 Q = A(I,M)
9042  FOR J = 0 TO N
9044 A(I,J) = A(I,J) - Q * A(M,J)

9046  NEXT J
9048  NEXT I
9050  NEXT M
9052  FOR I = 0 TO N
9054 A(I,I) = A(I,I) - 1
9056  NEXT I
9058  FOR I = 0 TO N
9060  FOR J = 0 TO N
9062  PRINT "A(";I + 1;",";J + 1;
     ")=";A(I,J),
9064  NEXT J
9066  NEXT I
9068  RETURN
9070  PRINT "ERROR": END
```

(5) 例题

① 求 $\begin{vmatrix} 1 & -2 & 0 \\ -1 & 3 & 2 \\ 1 & -1 & 4 \end{vmatrix}$ 的逆矩阵

② 主程序

```
10   DATA 3,1,-2,0,-1,3,2,1,-1,4
20   GOSUB 9000
30   END
```

③ 计算逆矩阵的子程序（AP-13）

④ 计算结果

```
URUN
A(1,1)=7            A(1,2)=4
```

```
A(1,3)=-2          A(2,1)=3
A(2,2)=2           A(2,3)=-1
A(3,1)=-1          A(3,2)=-.5
A(3,3)=.5
```

逆矩阵为：

$$\begin{bmatrix} 7 & 4 & -2 \\ 3 & 2 & -1 \\ -1 & -0.5 & 0.5 \end{bmatrix}$$

求矩阵乘积

(1) 目的

求两矩阵的乘积。

(2) 数学方法

m行 l列的矩阵 A与 l 行 n列的矩阵 B的乘积为m行 n列的矩阵C：

$$m\left\{\underbrace{\begin{bmatrix} a_{11} & a_{21} & \cdots a_{l1} \\ a_{21} & a_{22} & \cdots a_{2l} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots a_{ml} \end{bmatrix}}_{l}\right. \underbrace{\begin{bmatrix} b_{11} & b_{12} & \cdots b_{1n} \\ b_{21} & b_{22} & \cdots b_{2n} \\ \vdots & \vdots & \vdots \\ b_{l1} & b_{l2} & \cdots b_{ln} \end{bmatrix}}_{n} = \begin{bmatrix} c_{11} & c_{12} & \cdots c_{1n} \\ c_{21} & c_{22} & \cdots c_{2n} \\ \vdots & \vdots & \vdots \\ c_{m1} & c_{m2} & \cdots c_{mn} \end{bmatrix}$$

$$c_{ij} = \sum_{k=1}^{l} a_{ik} \cdot b_{kj} \qquad (i=1,2,\cdots, m; \quad j=1,2,\cdots, n)$$

(3) 子程序使用说明

① 简单变量

M——矩阵 A的行数；

L——矩阵 A的列数；

N——矩阵 B的列数；

B——矩阵 B顺序排列的各元素值。

② 数组

A（M－1， L－1）——矩阵A的各元素值；

C（M－1， N－1）——存放矩阵乘积的各元素值。

③ 数据排列顺序

M，L，A（M－1， L－1），N， B。

④ 计算结果

打印输出矩阵乘积的全部元素。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB 9000转本节子程序。

(4) 子程序

```
9000  REM MATRIX PRODUCT(AP-14)
9002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
9004  READ M,L
9006 M = M - 1
9008 L = L - 1
9010  DIM A(M,L)
9012  FOR I = 0 TO M
9014  FOR J = 0 TO L
9016  READ A(I,J)
9018  NEXT J
9020  NEXT I
9022  READ N
9024 N = N - 1
9026  DIM C(M,N)
```

```
9028  FOR I = 0 TO L
9030  FOR J = 0 TO N
9032  READ B
9034  FOR K = 0 TO M
9036 C(K,J) = C(K,J) + A(K,I) * B

9038  NEXT K
9040  NEXT J
9042  NEXT I
9044  FOR I = 0 TO M
9046  FOR J = 0 TO N
9048  PRINT "C(";I + 1;",";J + 1;
     ")="; FN P(C(I,J)),
9050  NEXT J
9052  NEXT I
9054  PRINT
9056  RETURN
```

(5）例题

① 求以下两矩阵之乘积:

$$\begin{bmatrix} 4 & 0 & -1 \\ -3 & 3 & 7 \\ -9 & 2 & 5 \\ 5 & -1 & 3 \end{bmatrix} \begin{bmatrix} -1 & 5 \\ -6 & -6 \\ 1 & 4 \end{bmatrix}$$

② 主程序

```
10   DATA 4,3
20   DATA 4,0,-1,-3,3,7,-9,2,5,5,-
      1,3
30   DATA 2
40   DATA -1,5,-6,-6,1,4
50   GOSUB 9000
60   END
```

③ 计算矩阵乘积的子程序（AP-14)

④ 计算结果

```
]RUN
C(1,1)=-5          C(1,2)=16
C(2,1)=-8          C(2,2)=-5
C(3,1)=2           C(3,2)=-37
C(4,1)=4           C(4,2)=43
```

即矩阵C为：

$$\begin{bmatrix} -5 & 16 \\ -8 & -5 \\ 2 & -37 \\ 4 & 43 \end{bmatrix}$$

求解线性代数方程组

(1) 目的

求解线性代数方程组。

(2) 数学方法

采用主元素消去法（无回代过程）求解。

设有方程组　AX=B

$$A=\begin{bmatrix} a_{00} & a_{01} & \cdots a_{0,n-1} \\ a_{10} & a_{11} & \cdots a_{1,n-1} \\ \vdots & \vdots & \vdots \quad \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots a_{n-1,n-1} \end{bmatrix}$$

$$X=\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

$$B=\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

首先找出整个系数矩阵A的模最大元素作为主元素，然后采用高斯消去方法求得方程组的解。

（3）子程序使用说明

① 简单变量

N——方程组中方程的个数。

② 数组

A(N-1, N)——按行存放系数矩阵A的元素，其中第n列存放方程组的右端项，用a_{0n}，a_{1n}，…$a_{n-1,n}$表示；

S(N)——存放方程组的解。

③ 数据排列顺序

N，A(N-1，N)。

④ 计算结果

打印输出方程组的解。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB 9000转本节子程序。

（4）子程序

```
9000  REM GAUSS(AP-15)
9002  DEF  FN P(Y) =  INT (Y * 10
      0 + 0.5) / 100
9004  READ N
9006  DIM A(N - 1,N),S(N),M(N - 1
      )
9008  FOR 1 = 0 TO N - 1
9010  FOR J = 0 TO N
9012  READ A(I,J)
9014  NEXT J
9016  NEXT I
9018  GOSUB 9022
9020  END
9022  FOR I = 0 TO N - 1
```

```
9024 P = I
9026 Q = 0
9028 E = A(I,0)
9030  FOR J = I TO N - 1
9032  FOR K = 0 TO N - 1
9034  IF  ABS (A(J,K)) <  =  ABS
     (E) THEN 9042
9036 E = A(J,K)
9038 Q = K
9040 P = J
9042  NEXT K
9044  NEXT J
9046  IF  ABS (E) > 10E - 10 THEN
     9052
9048  PRINT "NO UNIQUE SOLUTION"
9050  STOP
9052  IF P = I THEN 9064
9054  FOR K = 0 TO N
9056 S(K)  = A(I,K)
9058 A(I,K)  = A(P,K)
9060 A(P,K)  = S(K)
9062  NEXT K
9064  FOR J = 0 TO N - 1

9066  IF J = I THEN 9078
9068  IF A(J,Q)  = 0 THEN 9078
9070 R = A(J,Q)  / A(I,Q)
9072  FOR K = 0 TO N
9074 A(J,K)  = A(J,K)  - A(I,K)  * R

9076  NEXT K
9078  NEXT J
9080 M(I)  = Q
9082  NEXT I
9084  FOR I = 0 TO N - 1
9086 Q = M(I)
9088 S(Q)  = A(I,N)  / A(I,Q)
9090  NEXT I
9092  FOR Q = 0 TO N - 1
9094  PRINT "X";Q + 1;"="; FN P(S
```

```
        (Q)),
9096   NEXT Q
9098   RETURN
```

(5) 例题

① 设有方程组AX=B

式中：

$$A=\begin{bmatrix}2 & -1 & 1\\ 1 & 0 & 3\\ 5 & 2 & 1\end{bmatrix}$$

$$B=\begin{bmatrix}3\\ 10\\ 12\end{bmatrix}$$

解此方程组。

② 主程序

```
10   DATA 3,2,-1,1,3
20   DATA 1,0,3,10
30   DATA 5,2,1,12
40   GOSUB 9000
50   END
```

③ 解线性代数方程组的子程序(AP-15)

④ 计算结果

X1=1　X2=2　X3=3

4.1.4　方差分析

单因素方差分析

(1) 目的

如果只有一个因素分m个水平、类或等级做试验，每个水平重复不同次数，可通过方差分析检验这不同水平对试验结果的影响有无显著差异。

（2）数学方法

若单因素A有m个水平，每个水平重复次数分别为n_1，n_2，…，n_m，试验总次数为N，有

$$N=\sum_{i=1}^{m} n_i = n_1+n_2+\cdots+n_m$$

各水平各次试验值为x_{ij}

i为试验水平，$i=1,2,\cdots, m$；

j为每个试验水平重复序号，$j=1,2,\cdots,n_i$；

第i水平观测值的总和记为$x_{si}=\sum_{j=1}^{n_i} x_{ij}$；

X的总和计为$x_s=\sum_{i=1}^{m} x_{si}=\sum_{i=1}^{m}\sum_{j=1}^{n_i} x_{ij}$

$$P=\frac{1}{N}\left(\sum_{i=1}^{m}\sum_{j=1}^{n_i} x_{ij}\right)^2$$

A因素各水平合计平均平方和：

$$Q=\sum_{i=1}^{m}\frac{1}{n_j}\left(\sum_{j=1}^{n_i} x_{ij}\right)^2$$

个体平方和：

$$R=\sum_{i=1}^{m}\sum_{j=1}^{n_i} x_{ij}^2$$

各水平之间平方和：$S_A=Q-P$

误差平方和：$S_E=R-Q$

总平方和：$S_T=R-P$

它们的自由度分别为：

$$f_A = m-1,\ f_E = n-m,\ f_T = n-1$$

进行F检验：

$$F = \frac{Q-P}{R-Q} \cdot \frac{n-m}{m-1}$$

若$F(f_A, f_E) < F_{0.05}$，则认为这几个水平对于试验结果的影响没有显著差异；否则，差异显著。

(3) 子程序使用说明

① 简单变量

M——该因素的水平数目；

X——各水平样本数据；

FA——第一自由度；

FB——第二自由度；

F——统计量。

② 数组

N1(M－1)——各水平样本数目。

③ 数据排列顺序

M，N1(M－1)，X，…

④ 计算结果

打印输出在第一、二自由度下统计量F值。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB9000转本节子程序。

(4) 子程序

```
9000  REM SIN.ELE.ANA.(AP-16)
9002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
9004  READ M
9006 M = M - 1
```

```
9008  DIM N1(M),A(M)
9010  FOR I = 0 TO M
9012  READ N1(I)
9014  FOR J = 1 TO N1(I)
9016  READ X
9018 A(I) = A(I) + X
9020 R = R + X ^ 2
9022 N = N + 1
9024  NEXT J
9026 P = P + A(I)
9028 Q = Q + A(I) ^ 2 / N1(I)
9030  NEXT I
9032 P = P ^ 2 / N
9034 SA = Q - P:SB = R - Q:ST = R
      - P
9036 FA = M:FB = N - M - 1:FT = N
      - 1
9038 F = SA * FB / (SB * FA)
9040  PRINT "F(";FA;",";FB;")="; FN
     P(F)
9042  RETURN
```

(5) 例题

① 吉林省8月平均雨量从1951—1973年资料看有三年周期，用方差分析方法检验有无三年周期？

1951—1973 年雨量距平值为（设采用简化的数据）：5，－5，－2，3，－10，－3，9，－4，2，5，－1，－1，－5，8，5，8，－8，－2，3，－8，－3，2，－1

按三年周期排列，即为三个水平下的试验：

5	－5	－2
3	－10	－3
9	－4	2
5	－1	－1

-5	8	5
8	-8	-2
3	-8	-3
2	-1	

② 主程序

```
10   DATA 3
20   DATA 8,5,3,9,5,-5,8,3,2
30   DATA 8,-5,-10,-4,-1,8,-8,-8,-
      1
40   DATA  7,-2,-3,2,-1,5,-2,-3
50   GOSUB 9000
60   END
```

③ 单因素方差分析子程序（AP-16）

④ 计算结果

$F(2,20)=5.314$

查表得$F_{0.05}(2,20)=3.49$，故$F>F_{0.05}$，说明吉林省8月降水量有明显三年周期。

4.1.5　回归分析

相关系数

(1) 目的

求变量之间的相关系数。

(2) 数学方法

若有变量x_i($i=1,2,\cdots,M$)个，它们各有N次观测值，分别为x_{ij}（$j=1,2,\cdots,N$），另有变量y，也有N次观测值，分别为y_i（$j=1,2,\cdots,N$），则各个x_i与y的相关系数为：

$$R_i=\left(\sum_{j=1}^{N}x_{ij}y_j-\sum_{j=1}^{N}x_{ij}\sum_{j=1}^{N}y_j/N\right)\Big/$$

$$\left(\sqrt{\left(\sum_{j=1}^{N}x_{ij}^2-\left(\sum_{j=1}^{N}x_{ij}\right)^2/N\right)\left(\sum_{j=1}^{N}y_j^2-\left(\sum_{j=1}^{N}y_j\right)^2/N\right)}\right)$$

$$(i=1,2,\cdots,\ M)$$

（3） 子程序使用说明

① 简单变量

M——变量x的个数；

N——每个变量观测样本数目；

X——变量x观测值；

R——各个变量x与y的相关系数。

② 数组

Y(N－1)——变量y观测值。

③ 数据排列顺序

M, N, Y(N－1), X

④ 计算结果

打印输出每个变量x与y的相关系数值。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB9000转本节子程序。

（4） 子程序

```
9000   REM COR.COF.(AP-17)
9002   DEF  FN P(Y) =  INT (Y * 10
```

```
     0 + 0.5) / 100
9004  READ M,N:N = N - 1
9006  DIM Y(N)
9008  FOR I = 0 TO N
9010  READ Y(I)
9012 S1 = S1 + Y(I)
9014 S2 = S2 + Y(I) ^ 2
9016  NEXT I
9018 L3 = S2 - S1 ^ 2 / (N + 1)
9020  FOR I = 1 TO M
9022 A = 0:B = 0:C = 0
9024  FOR J = 0 TO N
9026  READ X
9028 A = A + X
9030 B = B + X ^ 2
9032 C = C + X * Y(J)
9034  NEXT J
9036 L1 = C - S1 * A / (N + 1)
9038 L2 = B - A ^ 2 / (N + 1)
9040 R = L1 /  SQR (L2 * L3)
9042  PRINT "R(";I;")="; FN P(R),

9044  NEXT I
9046  PRINT
9048  RETURN
```

(5) 例题

① 有两个变量x_1和x_2,观测值分别为:

x_1: 4, 11, 10, 7,8,7,6,12,7

x_2: 2, 3, 7, 6, 5,5,0,7,1

另有变量y,观测值为:

y: 15, 23, 24, 21, 22, 22,17, 26, 19

求: x_1,x_2与y的单相关系数。

② 主程序

```
10   DATA 2,9
20   DATA 15,23,24,21,22,22,17,26,
      19
30   DATA 4,11,10,7,8,7,6,12,7
40   DATA 2,3,7,6,5,5,0,7,1
50   GOSUB 9000
60   END
```

③ 计算相关系数子程序（AP-17）

④ 计算结果

```
RUN
R(1)=.92            R(2)=.81
```

一元线性回归

(1) 目的

有两个变量 x 和 y，用自变量 x 建立因变量 y 的线性回归方程。

(2) 数学方法

自变量 x，因变量 y，有N对观测，其线性回归方程是：

$$\hat{y} = a + bx$$

$$令 l_{xy} = \sum xy - \sum x \sum y / N$$

$$l_{xx} = \sum x^2 - (\sum x)^2 / N$$

$$l_{yy} = \sum y^2 - (\sum y)^2 / N$$

则 x 与 y 的相关系数 $R = l_{xy} / \sqrt{l_{xx} \cdot l_{yy}}$

$$b = l_{xy} / l_{xx}$$

$$a = \sum y / N - b \cdot \sum x / N$$

回归方程标准差$S=\sqrt{\dfrac{(1-R^2)l_{yy}}{N-2}}$

(3) 子程序使用说明

① 简单变量

N——变量观测样本对的数目；

X——变量x的样本值；

Y——变量y的样本值；

R——相关系数；

A，B——回归系数；

S——回归方程标准差。

② 数据排列顺序

N，X，Y，X，Y，…

③ 计算结果

打印输出相关系数，回归方程以及回归方程标准差。

④ 在主程序中按数据排列顺序给出数据后，由GOSUB 9000转本节子程序。

(4) 子程序

```
9000  REM LIN.REG.(AP-18)
9002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
9004  READ N
9006  FOR I = 1 TO N
9008  READ X,Y
9010 S1 = S1 + X
9012 S2 = S2 + Y
9014 S3 = S3 + X * Y
9016 S4 = S4 + X ^ 2
9018 S5 = S5 + Y ^ 2
9020  NEXT I
9022 L1 = S3 - S1 * S2 / N
9024 L2 = S4 - S1 ^ 2 / N
```

```
9026 L3 = S5 - S2 ^ 2 / N
9028 R = L1 /  SQR (L2 * L3)
9030 B = L1 / L2
9032 S =  SQR ((1 - R ^ 2) * L3 /
     (N - 2))
9034 A = (S2 - B * S1) / N
9036  PRINT "R="; FN P(R)
9038  PRINT "Y="; FN P(A);"+("; FN
     P(B);")*X"
9040  PRINT "S="; FN P(S)
9042  RETURN
```

(5) 例题

① 若有两组因子

x_i: 97, 47, 95, 84, 96, 113, 24, 16, 47, 37, 42, 14, 23, 35, 27, 85, 7, 113, 96, 68

y_i: 34, 17, 28, 25, 31, 52, 8, 6, 14, 9, 13, 3, 14, 10, 9, 36, 6, 24, 48, 12

求它们的相关系数及回归方程。

② 主程序

```
10   DATA 20
20   DATA 97,34,47,17,95,28,84,25,
      96,31
30   DATA 113,52,24,8,16,6,47,14,3
      7,9
40   DATA 42,13,14,3,23,14,35,10,2
      7,9
50   DATA 85,36,7,6,113,24,96,48,6
      8,12
60   GOSUB 9000
100   END
```

③ 求解一元线性回归方程子程序(AP-18)

④ 计算结果

```
URUN
R=.87
Y=-.25+(.35)*X
S=7.12
```

查表得$R_{0.001}=0.679$

因$R>R_{0.001}$故 x与 y 相关显著。

二元线性回归

(1) 目的

两个自变量x_1和x_2，一个因变量y，用x_1和x_2建立y的线性回归方程及其检验。

(2) 数学方法

自变量x_1，x_2，因变量y，它们的观测值为x_{1i}，x_{2i}，y_i，$i=1,2,\cdots$，N，N为样本数。若x_1和x_2与y之间有线性关系，则可以建立二元线性回归方程：

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i}$$

式中$\hat{y}_i$是因变量y_i的估计值，b_0，b_1，b_2是回归方程待定系数。

$$令 l_{11} = \sum x_{1i}^2 - (\sum x_{1i})^2/N$$

$$l_{12} = \sum x_{1i}x_{2i} - \sum x_{1i}\sum x_{2i}/N$$

$$l_{22} = \sum x_{2i}^2 - (\sum x_{2i})^2/N$$

$$l_{1y} = \sum x_{1i}y_i - \sum x_{1i}\sum y_i/N$$

$$l_{2y} = \sum x_{2i}y_i - \sum x_{2i}\sum y_i/N$$

$$l_{yy}=\sum y_i^2-(\sum y_i)^2/N$$

$$b_1=(l_{1y}l_{22}-l_{2y}l_{12})/(l_{11}l_{22}-l_{12}^2)$$

$$b_2=(l_{2y}l_{11}-l_{1y}l_{12})/(l_{11}l_{22}-l_{12}^2)$$

$$b_0=\bar{y}-b_1\bar{x}_1-b_2\bar{x}_2$$

x_1，x_2和 y 之间的单相关系数：

$$r_{12}=\frac{l_{12}}{\sqrt{l_{11}\cdot l_{22}}}$$

$$r_{1y}=\frac{l_{1y}}{\sqrt{l_{11}\cdot l_{yy}}}$$

$$r_{2y}=\frac{l_{2y}}{\sqrt{l_{22}\cdot l_{yy}}}$$

x_1和 x_2共同对因变量 y 的相关系数即复相关系数为：

$$R'=\sqrt{\frac{r_{1y}^2+r_{2y}^2-2r_{1y}r_{2y}r_{12}}{1-r_{12}^2}}$$

回归方程的标准差S：

$$S=\sqrt{\frac{l_{yy}-b_1l_{1y}-b_2l_{2y}}{N-3}}$$

统计量F为：

$$F=\left(\frac{R^2}{1-R^2}\right)\times\left(\frac{N-3}{2}\right)$$

第一自由度为2，第二自由度为N－3

(3) 子程序使用说明

① 简单变量

N——每组变量的样本个数；

B0, B1, B2——回归方程系数；

R——x_1与x_2的单相关系数；

RA——x_1与y的单相关系数；

RB——x_2与y的单相关系数；

RR——复相关系数；

F——统计量；

S——回归方程标准差。

② 数组

x（2，N－1）——存放自变量及因变量的数据，因变量数据放在最后。

③ 数据排列顺序

N，X（2，N－1）

④ 计算结果

打印输出回归方程表达式、单相关系数、复相关系数（Rx1x2.y）、统计量F和回归方程标准差。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB9000转本节子程序。

(4) 子程序

```
9000  REM TWO ELE.LIN.REG.(AP-19)

9002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
9004  READ N
9006 N = N - 1
9008  DIM X(2,N),S1(2),S2(2,2)
9010  FOR I = 0 TO 2
9012  FOR J = 0 TO N
9014  READ X(I,J)
9016  NEXT J
9018  NEXT I
9020  FOR I = 0 TO 2
9022  FOR K = 0 TO N
9024 S1(I) = S1(I) + X(I,K)
9026  NEXT K
```

```
9028  FOR J = I TO 2
9030  FOR K = 0 TO N
9032 S2(I,J) = S2(I,J) + X(I,K) *
     X(J,K)
9034  NEXT K
9036  NEXT J
9038  NEXT I
9040 L1 = S2(0,0) - S1(0) * S1(0)
      / (N + 1)
9042 L2 = S2(0,1) - S1(0) * S1(1)
      / (N + 1)
9044 L3 = S2(1,1) - S1(1) * S1(1)
      / (N + 1)
9046 L4 = S2(0,2) - S1(0) * S1(2)
      / (N + 1)
9048 L5 = S2(1,2) - S1(1) * S1(2)
      / (N + 1)
9050 L6 = S2(2,2) - S1(2) * S1(2)
      / (N + 1)
9052 C = L1 * L3 - L2 * L2
9054 B1 = (L4 * L3 - L5 * L2) / C
9056 B2 = (L5 * L1 - L4 * L2) / C

9058 B0 = (S1(2) - B1 * S1(0) - B
     2 * S1(1)) / (N + 1)
9060 R = L2 /  SQR (L1 * L3)
9062 RA = L4 /  SQR (L1 * L6)
9064 RB = L5 /  SQR (L3 * L6)
9066 RR =  SQR ((RA * RA + RB * R
     B - 2 * RA * RB * R) / (1 -
     R * R))
9068 S =  SQR ((L6 - B1 * L4 - B2
      * L5) / (N - 2))
9070 F = 0.5 * RR * RR * (N - 2) /
     (1 - RR * RR)
9072  PRINT "Y="; FN P(B0);"+("; FN
     P(B1);"*X1)+("; FN P(B2);"*X
     2)"
9074  PRINT "Rx1.x2="; FN P(R)
9076  PRINT "Rx1.y="; FN P(RA)
```

```
9078  PRINT "Rx2.y="; FN P(RB)
9080  PRINT "Rx1x2.y="; FN P(RR)
9082  PRINT "F(2,";N - 2;")="; FN
    P(F)
9084  PRINT "S="; FN P(S)
9086  RETURN
```

(5) 例题

① 若有两个自变量 x_1, x_2和因变量y

$x_1=(4, 11, 10, 7, 8, 7, 6, 12, 7)$

$x_2=(2, 3, 7, 6, 5, 5, 0, 7, 1)$

$y=(15, 23, 24, 21, 22, 22, 17, 26, 19)$

计算线性回归方程。

② 主程序

```
10   DATA 9
20   DATA 4,11,10,7,8,7,6,12,7
30   DATA 2,3,7,6,5,5,0,7,1
40   DATA 15,23,24,21,22,22,17,26,
      19
50   GOSUB 9000
60   END
```

③ 求解二元线性回归方程子程序(AP-19)

④ 计算结果

```
ÜRUN
Y=11.42+(.93*X1)+(.54*X2)
Rx1.x2=.59
Rx1.y=.92
Rx2.y=.81
Rx1x2.y=.98
F(2,6)=63.71
S=.85
```

查数理统计表$F_{0.01}(2, 6)=10.9$

由于$F(2,6)>F_{0.01}(2,6)$，所以回归方程显著。

多元线性回归

(1) 目的

如有m个自变量x_{1j}，x_{2j}，…，x_{ij}，…，x_{mj}，其中$i=1,2,\cdots,m$，$j=1,2,\cdots,N$，N为样本数；有因变量y_j，其中$j=1,2,\cdots,N$，若自变量和因变量有线性关系，可以有：

$$\hat{y}_j=b_0+b_1x_{1j}+b_2x_{2j}+\cdots+b_ix_{ij}+\cdots+b_mx_{mj}$$

式中$\hat{y}_j$是因变量y_j的估计值，$b_0,b_1,b_2,\cdots,b_i\cdots,b_m$是回归方程的待定系数。

(2) 数学方法

令：

$$l_{11}=\sum x_{1j}^2-(\sum x_{1j})^2/N$$

$$l_{12}=\sum x_{1j}x_{2j}-\sum x_{1j}\sum x_{2j}/N$$

$$\vdots$$

$$l_{1m}=\sum x_{1j}x_{mj}-\sum x_{1j}\sum x_{mj}/N$$

$$l_{22}=\sum x_{2j}^2-(\sum x_{2j})^2/N$$

$$\vdots$$

$$l_{2m}=\sum x_{2j}x_{mj}-\sum x_{2j}\sum x_{mj}/N$$

$$\vdots$$

$$l_{mm}=\sum x_{mj}^2-(\sum x_{mj})^2/N$$

$$l_{1y}=\sum x_{1j}y_j-\sum x_{1j}\sum y_j/N$$

$$l_{2y}=\sum x_{2j}y_j-\sum x_{2j}\sum y_j/N$$

$$\vdots$$

$$l_{my}=\sum x_{mj}y_j-\sum x_{mj}\sum y_j/N$$

$$l_{yy}=\sum y_j^2-(\sum y_j)^2/N$$

可得如下代数方程组：

$$\begin{aligned} b_1l_{11}+b_2l_{12}+\cdots+b_ml_{1m}&=l_{1y}\\ b_1l_{12}+b_2l_{22}+\cdots+b_ml_{2m}&=l_{2y}\\ &\vdots\\ b_1l_{1m}+b_2l_{2m}+\cdots+b_ml_{mm}&=l_{my}\end{aligned}$$

由高斯消去法求得b_1，b_2，…，b_m

$$b_0=\bar{y}-b_1\bar{x}_1-b_2\bar{x}_2-\cdots-b_m\bar{x}_m$$

各个因素之间的单相关系数是：

$$R_{12}=\frac{l_{12}}{\sqrt{l_{11}\cdot l_{22}}}$$

$$R_{13}=\frac{l_{13}}{\sqrt{l_{11}\cdot l_{33}}}$$

$$R_{1y}=\frac{l_{1y}}{\sqrt{l_{11}\cdot l_{yy}}}\qquad\text{等等。}$$

$$u=b_1l_{1y}+b_2l_{2y}+\cdots+b_ml_{my}$$

回归方程的复相关系数$R_{y.12\cdots m}$：

$$R_{y.12\cdots m}=\sqrt{u/l_{yy}}$$

统计量$F(f_A,\ f_B)$为：

$$F(f_A, f_B)=\frac{R^2_{y.12\cdots m}}{1-R^2_{y.12\cdots m}}\ \frac{N-m-1}{m}$$

第一自由度$f_A=m$，第二自由度$f_B=N-m-1$回归方程标准差S^*：

$$S^*=\sqrt{\frac{l_{yy}-u}{N-m-1}}$$

(3) 子程序使用说明

① 简单变量

M——自变量的个数；

N——每组变量观测样本的个数；

B0——回归方程系数；

RR——复相关系数；

F——统计量；

SS——回归方程标准差。

② 数组

X(M,N-1)——原始自变量及因变量数据；

S(M)——回归方程系数；

R(M,M)——变量之间单相关系数。

③ 数据排列顺序

M，N，X(M，N-1)

④ 计算结果

打印输出变量之间单相关系数，复回归方程表达式，复相关系数，统计量F和回归方程标准差SS。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB1000转本节子程序。

(4) 子程序

```
1000  REM MUL.ELE.REG.(AF-20)
```

```
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ M,N
1006 N = N - 1
1008  DIM X(M,N),S1(M),S2(M,M),L(
     M,M)
1010  DIM S(M),H(M - 1),R(M,M)
1012  FOR I = 0 TO M
1014  FOR J = 0 TO N
1016  READ X(I,J)
1018  NEXT J
1020  NEXT I
1022  FOR I = 0 TO M
1024  FOR J = 0 TO N
1026 S1(I) = S1(I) + X(I,J)
1028  NEXT J
1030  FOR J = I TO M
1032  FOR K = 0 TO N
1034 S2(I,J) = S2(I,J) + X(I,K) *
     X(J,K)
1036  NEXT K
1038  NEXT J
1040  NEXT I
1042  FOR I = 0 TO M
1044  FOR J = I TO M
1046 L(I,J) = S2(I,J) - S1(I) * S
     1(J) / (N + 1)
1048 L(J,I) = L(I,J)
1050  NEXT J
1052  NEXT I
1053  GOSUB 2000
1054  REM  GAUSS
1056  FOR I = 0 TO M - 1
1058 P = I
1060 Q = 0
1062 E = L(I,0)
1064  FOR J = I TO M - 1
```

```
1066  FOR K = 0 TO M - 1
1068  IF  ABS (L(J,K)) <  =  ABS
     (E) THEN 1074
1070 E = L(J,K)
1072 Q = K
1074 P = J
1076  NEXT K
1078  NEXT J
1080  IF  ABS (E) > 10E - 10 THEN
     1086
1082  PRINT "NO UNIQUE SOLUTION"
1084  STOP
1086  IF P = I THEN 1098
1088  FOR K = 0 TO M
1090 S(K)  = L(I,K)
1092 L(I,K)  = L(P,K)
1094 L(P,K)  = S(K)
1096  NEXT K
1098  FOR J = 0 TO M - 1
1100  IF J = I THEN 1112
1102  IF L(J,Q) = 0 THEN 1112
1104 R = L(J,Q) / L(I,Q)
1106  FOR K = 0 TO M
1108 L(J,K) = L(J,K) - L(I,K) * R

1110  NEXT K
1112  NEXT J
1114 H(I) = Q
1116  NEXT I
1118  FOR I = 0 TO M - 1
1120 Q = H(I)
1122 S(Q) = L(I,M) / L(I,Q)
1124  NEXT I
1126  FOR I = 0 TO M - 1
1128 SO = SO + S(I) * S1(I)
1130  NEXT I
1132 BO = (S1(M) - SO) / (N + 1)
```

```
1134  PRINT "Y=(";
1136  FOR I = 0 TO M - 1
1138  PRINT  FN P(S(I));")*X";I +
     1;"+(";
1140  NEXT I
1142  PRINT  FN P(BO);")"
1144  FOR I = 0 TO M - 1
1146 L(I,M) = S2(I,M) - S1(I) * S
  .  1(M) / (N + 1)
1148 U = U + S(I) * L(I,M)
1150  NEXT I
1152  PRINT
1154 RR =  SQR (U / L(M,M))
1156 F = RR * RR * (N - M) / ((1 -
     RR * RR) * M)
1158 SS =  SQR ((L(M,M) - U) / (N
     - M))
1160  PRINT "Ry.12...";M;"="; FN
     P(RR)
1162  PRINT "F(";M;",";N - M;")="
     ; FN P(F)
1164  PRINT "S="; FN P(SS)
1166  RETURN
2000  REM SIG.COR.COF.
2002  FOR I = 0 TO M - 1
2004  FOR J = I + 1 TO M
2006 R(I,J) = L(I,J) /  SQR (L(I,
     I) * L(J,J))
2008  PRINT "R(";I + 1;",";J + 1;
     ")="; FN P(R(I,J)),
2010  NEXT J
2012  NEXT I
2014  PRINT
2016  RETURN
```

(5) 例题

① 自变量x_1, x_2, x_3,因变量y的数值分别如下表示:

x_1: 0, 2, −1, −5, 6, 3, −10, 6, 5, −2,

3，1，7，-9，2，-3，0，4，-9，-5

x_2：-6，20，19，-16，5，-20，-10，13，29，6，-32，11，11，-4，3，4，-53，4，8，29

x_3：2，3，4，-2，1，-2，-2，2，2，5，3，-5，4，2，0，-6，-5，-5，-7，2

y：-8，4，7，-7，12，6，-14，4，9，3，-1，4，7，-3，5，-11，-8，-1，-11，6

求它们的复回归方程。

② 主程序

```
10   DATA 3,20
20   DATA 0,2,-1,-5,6,3,-10,6,5,-2
      ,3,1,7,-9,2,-3,0,4,-9,-5
30   DATA -6,20,19,-16,5,-20,-10,1
      3,29,6,-32,11,11,-4,3,4,-53,
      4,8,29
40   DATA 2,3,4,-2,1,-2,-2,2,2,5,3
      ,-5,4,2,0,-6,-5,-5,-7,2
45   DATA -8,4,7,-7,12,6,-14,4,9,3
      ,-1,4,7,-3,5,-11,-8,-1,-11,6

50   GOSUB 1000
60   END
```

③ 求解多元线性回归方程子程序（AP-20）

④ 计算结果

```
ÜRUN
R(1,2)=.105           R(1,3)=.291
R(1,4)=.684           R(2,3)=.313
R(2,4)=.49            R(3,4)=.565

Y=(.824)*X1+(.129)*X2+(.599)*X3+(.341)

Ry.12...3=.847
```

```
F(3,16)=13.535
S=4.413
```

R(1,4)即为x_1与y的单相关系数；R(2,4)为x_2与y的单相关系数；R(3,4)为x_3与y的单相关系数。

由数理统计表查得$F_{0.01}(3,16)=5.3$，故$F>F_{0.01}$，所得回归方程是有效的。

逐步回归

(1) 目的

逐步回归是一种回归分析方法，它可从一组自变量中选取若干个最有效的自变量构成因变量的回归方程。

(2) 数学方法

① 基本思路

从一组自变量中根据自变量重要性的大小，每步选择一个重要变量进入回归方程。一般地说，第l步是在未选的量中，选一个这样的量，它与已选量组成的l元回归方程将比其它任一个量与已选量组成的l元方程有更大的平方和。另外对每步即将入选的变量作显著性检验，以保证选入回归方程的变量是重要的，如果检验不显著便停止挑选变量。本程序不仅考虑到按变量的贡献大小逐一选出重要变量，而且当较早选入回归方程的某个变量随着其后另一些变量的选入而失去原有的重要性时，及时地从回归方程中剔除出去。

在逐步回归计算中实际上是用无回代过程的消去法解正规方程，在解方程的同时进行重要变量的挑选。在消去过程中可以得到一系列过渡性回归方程，且在求解过程中同时求逆矩阵。

② 计算步骤

a) 建立正规方程

计算均值

$$\bar{x}_i = \frac{1}{N}\sum_k x_{ki} \qquad i=1,\ 2,\cdots,m,\ y*$$

计算离差阵

$$S_{ij}=S_{ji}=\sum_k (x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j)$$

$$i,j=1,\ 2,\ \cdots,m,\ \ y*$$

建立正规方程

$$\begin{cases} S_{11}b_1+S_{12}b_2+\cdots+S_{1m}b_m=S_{1y} \\ S_{21}b_1+S_{22}b_2+\cdots+S_{2m}b_m=S_{2y} \\ \quad\vdots \\ S_{m1}b_1+S_{m2}b_2+\cdots+S_{mm}b_m=S_{my} \end{cases}$$

为使计算有更好的数字效果，可把正规方程改为：

$$\begin{cases} r_{11}\tilde{b}_1+r_{12}\tilde{b}_2+\cdots+r_{1m}\tilde{b}_m=r_{1y} \\ r_{21}\tilde{b}_1+r_{22}\tilde{b}_2+\cdots+r_{2m}\tilde{b}_m=r_{2y} \\ \vdots \\ r_{m1}\tilde{b}_1+r_{m2}\tilde{b}_2+\cdots+r_{mm}\tilde{b}_m=r_{my} \end{cases}$$

* 这里把 y 看作是形式为x_y的量

其中r_{ij}是相关系数

$$r_{ij}=\frac{S_{ij}}{\sqrt{S_{ii}}\cdot\sqrt{S_{jj}}}$$

$$i,j=1,\ 2,\ \cdots,\ m,\ y$$

新方程的解$\tilde{b}_i$与原正规方程的解b_i有如下关系：

$$b_i=\tilde{b}_i\sqrt{S_{yy}}/\sqrt{S_{ii}} \qquad i=1,\ 2,\ \cdots,\ m$$

相关矩阵(r_{ij})的逆矩阵$(\tilde{C}_{ij})$与离差矩阵(S_{ij})的逆矩阵(C_{ij})有如下关系：

$$C_{ij}=\frac{\tilde{C}_{ij}}{\sqrt{S_{ii}}\cdot\sqrt{S_{jj}}}$$

新的残差平方和($\tilde{Q}$)、回归平方和($\tilde{U}$)、贡献($\tilde{V}_i$) 等值都与原值相差一个比例因子S_{yy}

b) 逐步计算

计算的每一步先考虑变量的剔除，仅当不需剔除时再考虑引入变量，在既不能剔除也无法引入的情况下结束逐步计算。

c) 说明

在各个变量贡献的显著性检验中，若临界值F_α取得足够小（特别令$F_\alpha=0$）则全部变量都被选中。如果希望多选些变量，F_α可取较小的值，反之取较大值，但对于N较小时，变量个数不应选得过大，否则不可靠。本程序把临界值F_α区分为F_1和F_2，F_1用于引入，F_2用于剔除。显然要求$F_1 \geqslant F_2$，否则可能导致某个变量进进出出，循环不止。本程序用$F_1+0.1\times10^{-7}$ 代替原来给出的F_1。

为保证计算顺利，计算中如遇到$r_{ii}<0.1\times10^{-7}$，则x_i不考虑引进。

本程序由“概率统计计算”一书（科学出版社）上的逐步回归程序（BCY语言）改写而成（BASIC语言），主要有以下几点作了改动：

删掉了计算残差和偏相关的内容，因此只用一维数组装观测数据，并对计算均值和离差矩阵的有关程序部分的计算方法作了相应的改动，这样不必考虑观测次数N太多会影响内存，故资料长度可以不限。

控制打印指示的变量P_{123}的基本内容保持不变，既然删掉了计算残差的内容，所以$P_2=1$和$P_3=4$暂时不用。如

想增加计算残差的内容，则装观测数据的数组的维数，计算均值和离差部分以及控制打印指示的部分都应作相应改动。

所有的打印过程全部改写。

(3) 子程序使用说明

① 简单变量

M——自变量的个数；

N——观测次数；

S(显示SYY=)——因变量y的距平平方和；

L(显示L=)——选出的重要变量个数；

P(显示STEP=)——逐步回归执行的步数；

U(显示K12=)——引入或剔除变量的序号；

R(显示RY12M)——复相关系数；

F(显示F12=)——统计量；

Z(显示SY=)——y的剩余标准差；

W(显示F=)——回归方差与剩余方差之比值，即对R_{y12m}的检验值；

F1——引入变量时所取临界值；

F2——剔除变量时所取临界值，要求F1≥F2；

E——指示变量。若计算顺利，则E=0；否则E<0，其中E=-1表示某变量为常数，无法计算相关系数；E=-2，表示在当前的临界值F_1，F_2下不能选出重要变量。

PP(显示P123=)——选择打印输出的项目，由三位数字组成，记为$P_1P_2P_3$，若PP=0，则不作任何打印否则：

P1=0：无打印。

1：打印最终的BI(M)及L，R_{y12M}，F，S_y。

2：除包括P1=1的内容外，加印S_{yy}、均值（MX(M)），标准差(VX(M))，x_i与y的简单相关系数（VYX

(M)), 以及最终的TI(M－1)。

3: 除包括P1＝2加印S_{ij}与r_{ij}(即R矩的内容外，阵:下三角阵为S_{ij}，上三角阵为r_{ij})。

4: 除包括P1＝3的内容外,加印最终的r_{ij}(即逆阵$\widetilde{C}_{ij}$)。

P2＝0:无打印。

1: (暂时不用)。

P3＝0:无打印。

1: 逐步印S_{tep}, K_{12}, F_{12}, L, R_{y12m}, F, S_y。其中S_{tep}为步数，K_{12}为引入或剔除变量的序号($K_{12}<0$为剔除)，F_{12}为F统计量的值。

2: 除包括P3＝1的内容外,并加印每步的BI,TI。

3: 除包括P3＝2的内容外，加印每步的r_{ij}。

4: (暂时不用)。

其中参数M，N，XY，F_1，F_2，P123是应由使用者给出基本数据的。

② 数组

XY(M)——存放观测数据，最后一组是因变量;

BI(M)——存放算出的回归系数b_0, b_1, b_m的数组;

TI(M－1)——存放各个变量t检验值的数组;

MX(M)——存放各个变量的平均值;

VX(M)——存放各个变量的标准差;

VYX(M)——x_i与y的简单相关系数;

R(M，M)——存放R矩阵，下三角阵为S_{ij}，上三角阵为r_{ij}。

③ 数据排列顺序

M，N，XY(M)。

④ 计算结果

根据给显示标记 P 123 的不同数据，分别打印输出不同组合的结果，详见对简单变量PP的说明。

⑤ 在主程序的数据语句中，按数据排列顺序给出具体数据后，由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM STEP REG.(AP-21)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ M,N
1006  DIM R(M,M),XY(M),VX(M),SX(M
     ),VYX(M),MX(M)
1008  FOR J = 0 TO M
1010  READ XY(J): NEXT J
1012  FOR I = 0 TO M
1014 SX(I) = SX(I) + XY(I)
1016  FOR J = 0 TO I
1018 R(I,J) = R(I,J) + XY(J) * XY
     (I)
1020  NEXT J: NEXT I
1022 L = L + 1
1024  IF L < N THEN 1008
1026  FOR I = 0 TO M
1028 MX(I) = SX(I) / N
1030  FOR J = 0 TO I
1032 B = R(I,J)
1034 R(I,J) = B - N * MX(I) * MX(
     J)
1036  NEXT J: NEXT I
1038  FOR I = 0 TO M
1040 S = R(I,I)
1042  IF S = 0 THEN 1046
1044 VX(I) =  SQR (S): NEXT I: GOTO
     1048
1046 E =  - 1: PRINT "E=";E: STOP

1048  FOR I = 1 TO M
```

```
1050 D = VX(I)
1052  FOR J = 0 TO I - 1
1054 R(J,I) = R(I,J) / (D * VX(J)
     )
1056  NEXT J: NEXT I
1058 D =  SQR (1 / (N - 1))
1060  FOR I = 0 TO M
1062 VX(I) = VX(I) * D
1064 VYX(I) = R(I,M): NEXT I
1066  INPUT "P123=";PP
1068 P1 =  INT (PP / 100)
1070 D = PP - 100 * P1
1072 P2 =  INT (D / 10)
1074 P3 = D - 10 * P2
1076  IF P1 < 2 THEN 1080
1078 VYX(M) = 1: GOTO 1086
1080  IF P1 < 3 THEN 1108
1082  GOSUB 1098
1084  GOTO 1108
1086  PRINT "SYY="; FN P(S)
1088  FOR J = 0 TO M
1090  PRINT "MX(";J + 1;")="; FN
     P(MX(J)),
1092  PRINT "VX(";J + 1;")="; FN
     P(VX(J)),
1094  PRINT "VYX(";J + 1;")="; FN
     P(VYX(J)),
1096  NEXT J: PRINT : GOTO 1080
1098  FOR I = 0 TO M
1100  FOR J = 0 TO M
1102  PRINT "R(";I + 1;",";J + 1;
     ")="; FN P(R(I,J)),
1104  NEXT J: PRINT : NEXT I
1106  RETURN
1108  FOR I = 0 TO M
1110 R(I,I) = 1:VYX(I) = VX(M) /
     VX(I)
1112  IF I = M THEN 1120
```

```
1114  FOR J = I + 1 TO M
1116 R(J,I) = R(I,J)
1118  NEXT J
1120  NEXT I
1122  STOP
1124 L = 0:P = 0:Q = 1
1126  DIM TI(M - 1),BI(M)
1128  INPUT "F1=";A: INPUT "F2=";
     B
1130 P = P + 1:H = 0:G = 10
1132  FOR I = 0 TO M - 1
1134 TI(I) = 0:D = R(I,I)
1136  IF D < 1E - 08 THEN 1154
1138 V = (R(M,I) / D) * R(I,M)
1140  IF V < 0 THEN 1146
1142  IF V <  = H THEN 1154
1144 H = V:T = I: GOTO 1154
1146 TI(I) = D
1148  IF G >  - V THEN 1152
1150  GOTO 1154
1152 G =  - V:C = I: GOTO 1154
1154  NEXT I
1156  IF L = 0 THEN 1166
1158  IF F3 < 2 THEN 1162
1160 Y = 2: GOSUB 1240
1162  IF F3 < 3 THEN 1166
1164  GOSUB 1098
1166 F = (N - L - 1) * G / Q
1168  IF F < B THEN 1176
1170 F = (N - L - 2) * H / (Q - H
     )
1172  IF F <  = (A + 1E - 08) THEN
     1216
1174 L = L + 1:K = T:U = K: GOTO
     1178
1176 L = L - 1:K = C:U =  - K
1178 D = 1 / R(K,K):R(K,K) = 1
```

```
1180  FOR J = 0 TO M
1182 R(K,J) = R(K,J) * D: NEXT J
1184  FOR I = 0 TO M
1186  IF I = K THEN 1196
1188 D = R(I,K):R(I,K) = 0
1190  FOR J = 0 TO M
1192 R(I,J) = R(I,J) - D * R(K,J)
1194  NEXT J
1196  NEXT I
1198 Q = R(M,M):R =  SQR (1 - Q)
1200 W = (N - L - 1) * (1 - Q) /
     (L * Q)
1202 Z =  SQR (S * Q / (N - L - 1
     ))
1204  IF P3 < 1 THEN 1214
1206  PRINT "STEP=";P; SPC( 5);"K
     12=";U
1208  PRINT "F12="; FN P(F); SPC(
     5);"L=";L
1210  PRINT "RY12M="; FN P(R); SPC(
     5);"F="; FN P(W)
1212  PRINT "SY="; FN P(Z)
1214  GOTO 1130
1216  IF L = 0 THEN E =  - 2: GOTO
     1236
1218 E = 0
1220  IF P3 < 2 THEN 1224
1222  GOTO 1230
1224  IF P1 = 0 THEN 1230
1226  PRINT "L=";L; SPC( 5);"RY12
     M="; FN P(R); SPC( 5);"F="; FN
     P(W)
1228  PRINT "SY="; FN P(Z):Y = P1
     : GOSUB 1240
1230  IF P1 < 4 THEN 1238
1232  IF P3 < 3 THEN  GOSUB 1098
1234  GOTO 1238
```

```
1236  PRINT "E=";E
1238  RETURN
1240 D = 0
1242  FOR I = 0 TO M - 1
1244  IF TI(I) = 0 THEN 1254
1246 X = R(I,M):BI(I + 1) = VYX(I
     ) * X
1248 D = BI(I + 1) * MX(I) + D
1250 TI(I) = X /  SQR (TI(I) * Q /
     (N - L - 1))
1252  GOTO 1256
1254 BI(I + 1) = 0
1256  NEXT I
1258 BI(0) = MX(M) - D
1260  IF Y = 0 THEN 1282
1262  IF Y < 2 THEN 1266
1264  GOTO 1274
1266  FOR I = 0 TO M
1268  PRINT "B(";I;")="; FN P(BI(
     I)),: NEXT I
1270  PRINT
1272  GOTO 1282
1274  FOR I = 0 TO M: PRINT "B(";
     I;")="; FN P(BI(I)),: NEXT I
     : PRINT
1276  FOR I = 0 TO M - 1: PRINT "
     T(";I + 1;")="; FN P(TI(I)),

1278  NEXT I
1280  PRINT
1282  RETURN
```

(5) 例题

① 已知因变量y随4个自变量x_1，x_2，x_3，x_4而变化，并有32次观测数据，见下表：

表 4.1

序号	自变量				因变量	计算值
	x_1	x_2	x_3	x_4	y	$\hat{y}$
1	13	7	26	19	11.5	10.9966
2	15	11	40	34	19.8	19.5262
3	21	8	29	17	13.7	14.0475
4	19	12	15	33	21.6	21.0516
5	27	11	13	27	22.3	22.0982
6	32	10	21	15	19.1	18.6183
7	17	8	18	16	11.7	11.5199
8	26	10	35	23	19.4	19.5872
9	14	6	14	18	10.6	11.0021
10	28	13	21	34	25.5	26.1124
11	19	9	13	29	18.7	19.0473
12	12	10	19	38	19.3	20.0107
13	23	8	25	17	15.6	15.0608
14	28	11	33	32	24.7	25.1103
15	21	9	18	19	15.3	15.0497
16	35	14	24	34	29.8	29.6589
17	16	6	19	14	10.2	10.0110
18	24	10	32	26	19.8	20.0772
19	22	11	39	38	25.3	25.0770
20	10	7	17	20	9.7	9.9778
21	18	8	34	22	14.8	15.0330
22	29	11	28	21	20.7	20.1049
23	18	11	16	32	19.6	20.0439
24	16	10	15	34	20.4	20.0328
25	18	7	23	14	11.1	11.0243
26	23	11	29	29	20.7	21.0738

续表

序号	自变量				因变量	计算值
	x_1	x_2	x_3	x_4	y	$\hat{y}$
27	25	13	41	40	28.9	27.5991
28	32	9	12	15	18.3	18.6183
29	36	11	37	18	21.5	22.1481
30	31	9	25	14	17.7	17.6106
31	29	13	14	38	28.3	28.6234
32	18	10	11	35	21.6	21.5472

要求用逐步回归方法从这4个变量中选出起重要作用的因子，建立y的回归方程。

② 主程序

```
5   DATA 4,32
10   DATA 13,7,26,19,11.5,15,11,40
      ,34,19.8
15   DATA 21,8,29,17,13.7,19,12,15
      ,33,21.6
20   DATA 27,11,13,27,22.3,32,10,2
      1,15,19.1
25   DATA 17,8,18,16,11.7,26,10,35
      ,23,19.4
30   DATA 14,6,14,18,10.6,28,13,21
      ,34,25.5
35   DATA 19,9,13,29,18.7,12,10,19
      ,38,19.3
40   DATA 23,8,25,17,15.6,28,11,33
      ,32,24.7
45   DATA 21,9,18,19,15.3,35,14,24
      ,34,29.8
50   DATA 16,6,19,14,10.2,24,10,32
      ,26,19.8
55   DATA 22,11,39,38,25.3,10,7,17
```

```
      ,20,9.7
60  DATA 18,8,34,22,14.8,29,11,28
      ,21,20.7
65  DATA 18,11,16,32,19.6,16,10,1
      5,34,20.3
70  DATA 18,7,23,14,11.1,23,11,29
      ,29,20.7
75  DATA 25,13,41,40,28.9,32,9,12
      ,15,18.3
80  DATA 36,11,37,18,21.5,31,9,25
      ,14,17.7
85  DATA 29,13,14,38,28.3,18,10,1
      1,35,21.6
90  GOSUB 1000
95  END
```

③ 计算逐步回归子程序（AP-21）

④ 计算结果

RUN主程序，显示P123＝　，若键入100(即选择F1＝1，P2＝0，P3＝0)，则运行片刻后暂停，键入CONT；继续运行，显示F1＝　，若键入2.5则又显示F2＝　，又键入2.5，则运行片刻后，打印出计算结果，如下：

```
URUN
P123=100

BREAK IN 1122
UCONT
F1=2.5
F2=2.5
L=2         RY12M=.997        F=2178.075
SY=.462
B(0)=-5.11          B(1)=.507
B(2)=0              B(3)=0              B(4)=.501
```

由逐步回归计算（F1＝F2＝2.5）选出重要的变量是x_1

及 x_4，相应的回归系数是：

$b_1 = 0.507 \qquad b_4 = 0.501$

常数项　$b_0 = -5.11$

即回归方程是：

$$y = b_0 + b_1 x_1 + b_4 x_4 = -5.11 + 0.51x_1 + 0.50x_4$$

若RUN主程序，显示Y 123=　，键入400(即选择P1 = 4，P2 = 0，P3 = 0)，则运行片刻后暂停，键入CONT，继续运行，显示F1=　，若键入2.5，则又显示F2=　，又键入2.5，则运行片刻后，打印出计算结果的另一组合，如下：

```
ÜRUN
P123=400
SYY=935.265
MX(1)=22.344       VX(1)=6.926
VYX(1)=.604        MX(2)=9.813
VX(2)=2.07         VYX(2)=.957
MX(3)=23.625       VX(3)=9.058
VYX(3)=.228        MX(4)=25.469
VX(4)=8.696        VYX(4)=.766
MX(5)=18.972       VX(5)=5.493
VYX(5)=1
R(1,1)=1487.219    R(1,2)=.567
R(1,3)=.21         R(1,4)=-.043
R(1,5)=.604
R(2,1)=252.063     R(2,2)=132.875
R(2,3)=.208        R(2,4)=.741
R(2,5)=.957
R(3,1)=408.125     R(3,2)=120.75
R(3,3)=2543.5      R(3,4)=.1
R(3,5)=.228
R(4,1)=-81.156     R(4,2)=413.813
R(4,3)=244.625     R(4,4)=2343.969
R(4,5)=.766
R(5,1)=712.809     R(5,2)=337.231
R(5,3)=351.363     R(5,4)=1133.422
R(5,5)=935.265
```

```
BREAK IN 1122
UCONT
F1=2.5
F2=2.5
L=2        RY12M=.997        F=2178.075
SY=.462
B(0)=-5.11          B(1)=.507
B(2)=0              B(3)=0              B(4)=.501
T(1)=42.266         T(2)=0              T(3)=0
T(4)=52.481
```

```
R(1,1)=1.002      R(1,2)=.6
R(1,3)=.215       R(1,4)=.044
R(1,5)=.639
R(2,1)=-.6        R(2,2)=.09
R(2,3)=5E-03      R(2,4)=-.768
R(2,5)=6E-03
R(3,1)=-.215      R(3,2)=5E-03
R(3,3)=.944       R(3,4)=-.11
R(3,5)=.014
R(4,1)=.044       R(4,2)=.768
R(4,3)=.11        R(4,4)=1.002
R(4,5)=.793
R(5,1)=-.639      R(5,2)=6E-03
R(5,3)=.014       R(5,4)=-.793
R(5,5)=7E-03
```

所得出的回归方程与上次运行一样，增加了其它显示内容。

4.1.6 判别分析

逐步判别

(1) 目的

逐步判别是判别分析的一种，它按逐步回归的思想从大量因子中挑选若干必要的、最好组合的几个因子建立判别函数，由此可以判别某一个个体属于哪一类。本程序用于多级

逐步判别的计算。

(2) 数学方法

首先计算总离差矩阵T和类内离差矩阵W，然后每一步挑选一个最好因子检验，通过则选入，然后计算剔除，直到不再选入不再剔除为止。如一共选了r个因子，则第r + 1步就要计算判别系数，在计算判别系数时，采用如下的判别函数：

$$y_j = \ln q_j + C_{0j} + C_{1j} x_1 + C_{2j} x_2 + \cdots + C_{mj} x_m$$

$j = 1, 2, \cdots, G$ 表示类

q_j为先验概率，在各类次数相差不大时这一项往往略去，成为：

$$y_j = C_{0j} + C_{1j} x_1 + \cdots C_{mj} x_m$$

$j = 1, 2, \cdots, G$

比较每类的y_j，选其中最大的就判为这一类[1)]

$$y_j^* = \max_{1 \leqslant j \leqslant G} \left\{ y_j \right\}$$

(3) 子程序使用说明

① 简单变量

G——分类数；

M——自变量的个数（即因子数）；

N——每个变量观测次数（即每个变量样本数）；

F1——用作选入因子的F检验临界值F_α；

F2——用作剔除因子的F检验临界值F_α；

L——最终选中的变量个数；

1）详见参考文献[8]。

U——最终的Wilks的数值；

X2——U的近似检验χ^2值；

R 12——该步引入或剔除的变量序号，正值为引入，负值为剔除。

P1,P2，P3——用于控制打印输出的变量，程序运行中键盘输入，其定义如下：

P1＝0：不打印。

1：打印各类的样本数NG(G),最终的C(G,M), DM(G+1，G＋1)，L，U，χ^2，D2HG ((G－1)*G/2)，FHG((G－1)*G/2)。

2：除打印1的内容外，还打印分类平均值及总平均值MX(G＋1,M)。

3：除打印2的内容外,还打印初始的W(M,M),T (M，M)。

4：除打印3的内容外,还打印逐步计算后的W(M, M),T(M，M)。

P2＝0：不打印。

1：打印分类情况明细表。

P3＝0：不打印。

1：逐步打印STEP(步数),R12,F，L,U,χ^2.

2：除打印1的内容外,还逐步打印C(G,M),DM (G＋1,G＋1)，D2HG((G－1)*G/2)， FHG((G－1)*G/2)。

3：除打印2的内容外,还逐步打印W(M，M)，T (M,M)。

4：除打印3的内容外,还逐步打印分类情况明细表。

② 数组

Q(G)——存放先验概率，当Q(0)=0时，自动用样本频率代替先验概率；当Q(0)=1时，设为先验概率相等；若Q(0)为其它值，则从数据语句中读给定的先验概率值；

XY(N, M+1)——存放N次观测数据，前M列存放自变量，第M+1列存放原分类号；

C(G, M)——存放判别系数，未选变量的系数为零；

W(M, M)——存放组内离差平方和阵；

T(M, M)——存放总离差平方和阵；

D2HG((G-1)*G/2)——存放D^2ef, f<e=2, 3, …, G，它表示对第e和f两个母体的判别效果；

FHG((G-1)*G/2)——存放D^2ef的F检验值；

DM(G+1,G+1)——存放判别矩阵，同一行的为计算分类同一类；同一列的为原分类同一类，最后一行和一列为小计；

MX(G+1,M)——存放各变量的分类平均及总平均值；

NG(G)——各类的样本数目。

③ 数据排列顺序

G, M, N, F1,F2,XY(N, M+1)，根据预先的要求存放先验概率值。

④ 计算结果

根据给予显示标记"P1,P2,P3="的不同数据，分别打印输出不同组合的结果，详见对简单变量P1, P2,P3的说明。

⑤ 在主程序的数据语句中，按数据排列顺序给出具体数据后，由GOSUB1000转本节子程序。

(4) 子程序

```
1000  REM STEP JUD.(AP-22)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  INPUT "P1,P2,P3=";P1,P2,P3
1006  READ G,M,N,F1,F2
1008  DIM Q(G),XY(N,M + 1),C(G,M)
     ,W(M,M)
1010  DIM T(M,M),MX(G + 1,M),PR(G
     ),NG(G),X(M),DM(G + 1,G + 1)

1012  DIM D2HG((G - 1) * G / 2),F
     HG((G - 1) * G / 2),LQQ(G)
1014  INPUT "Q(0)=";Q(0)
1016  FOR I = 1 TO M + 1: FOR J =
     1 TO N
1018  READ XY(J,I): NEXT J: NEXT
     I
1020  IF Q(0) = 0 OR Q(0) = 1 THEN
     1024
1022  FOR I = 1 TO G: READ Q(I): NEXT
     I
1024 G1 = G + 1:M1 = M + 1
1026  FOR I = 1 TO G + 1: FOR J =
     1 TO M:MX(I,J) = 0: NEXT J: NEXT
     I
1028  FOR I = 1 TO G:NG(I) = 0: NEXT
     I
1030  FOR K = 1 TO N
1032 H = XY(K,M1)
1034  IF (H - 1) * (H - G) <  = 0
      THEN 1038
1036  STOP
1038 NG(H) = NG(H) + 1
1040  FOR I = 1 TO M
1042 MX(H,I) = MX(H,I) + XY(K,I)
1044  NEXT I: NEXT K
1046  FOR I = 1 TO M
1048 D = 0
```

```
1050  FOR H = 1 TO G
1052 D = D + MX(H,I)
1054 MX(H,I) = MX(H,I) / NG(H)
1056  NEXT H
1058 MX(G1,I) = D / N
1060  NEXT I
1062  IF P1 = 0 THEN 1082
1064  IF P1 < 2 THEN 1080
1066  PRINT : PRINT : PRINT : PRINT

1068  PRINT "   ";"MX(I,J)="
1070  FOR I = 1 TO G + 1: PRINT :
      PRINT "I=";I;":";:K = 0
1072  FOR J = 1 TO M:Z = MX(I,J)
1074  PRINT  FN P(Z),:K = K + 1
1076  IF K = 4 THEN K = 0: PRINT

1078  NEXT J: PRINT : NEXT I
1080  PRINT : PRINT : FOR I = 1 TO
     G: PRINT "NG(";I;")=";NG(I):
      NEXT I: PRINT
1082  FOR K = 1 TO N
1084  FOR I = 1 TO M
1086 D = XY(K,I) - MX(G1,I)
1088 X(I) = D
1090  FOR J = 1 TO I
1092 T(I,J) = T(I,J) + D * X(J)
1094  NEXT J
1096  NEXT I: NEXT K
1098  FOR H = 1 TO G
1100 D = NG(H)
1102  FOR I = 1 TO M
1104 D1 = MX(H,I) - MX(G1,I)
1106 X(I) = D * D1
1108  FOR J = 1 TO I
1110 W(I,J) = W(I,J) + D1 * X(J)
1112  NEXT J
1114  NEXT I: NEXT H
```

```
1116  FOR I = 1 TO M
1118  FOR J = 1 TO I
1120 W(I,J) = T(I,J) - W(I,J)
1122 W(J,I) = W(I,J)
1124 T(J,I) = T(I,J)
1126  NEXT J: NEXT I
1128  IF P1 < 3 THEN 1132
1130  GOSUB 1352
1132 D1 = Q(0)
1134  FOR H = 1 TO G
1136  IF D1 = 0 THEN D = NG(H) /
     N: GOTO 1142
1138  IF D1 = 1 THEN D = 1 / G: GOTO
     1142
1140 D = Q(H)
1142 LQQ(H) =  LOG (D)
1144  NEXT H
1146  REM STEP CAL.
1148 STE = 0:L = 0:EPS = 0.000001
     :U = 1
1150  FOR I = 1 TO M:X(I) = 0: NEXT
     I
1152 STE = STE + 1:UMI = 1000:UAX
      = EPS
1154  FOR I = 1 TO M
1156  IF X(I) <  > 0 THEN 1180
1158  IF T(I,I) < EPS THEN 1180
1160 UI = W(I,I) / T(I,I)
1162  IF UI >  = EPS THEN 1176
1164  PRINT "I=";I
1166  PRINT "UI=";UI
1168  PRINT "X="
1170  FOR J = 1 TO M: PRINT "X(";
     J;")=";X(J): NEXT J
1172 UI = EPS
1174  IF W(I,I) = 0 THEN 1276
```

```
1176  IF UI < UMI THEN UMI = UI:I
     MI = I
1178  GOTO 1186
1180 UI = T(I,I) / W(I,I)
1182  IF UI <  = UAX THEN 1186
1184 UAX = UI:IAX = I
1186  NEXT I
1188 F = (1 - UAX) * (N - L - G +
     1) / (UAX * (G - 1))
1190  IF F < F2 THEN 1200
1192 F = (1 - UMI) * (N - L - G) /
     (UMI * (G - 1))
1194  IF F <  = (F1 + EPS) THEN 1
     254
1196 L = L + 1:R = IMI:R12 = R:X(
     R) = 1
1198  GOTO 1202
1200 L = L - 1:R = IAX:R12 =  - R
     :X(R) = 0
1202 U = U * (W(R,R) / T(R,R))
1204 X2 =  - (N - 1 - (L + G) / 2
     ) *  LOG (U)
1206  IF P3 = 0 THEN 1214
1208  PRINT : PRINT : PRINT
1210  PRINT "STEP=";STE;"R12=";R1
     2;"F="; FN P(F)
1212  PRINT "L=";L;"U="; FN P(U);
     "X2="; FN P(X2): PRINT : PRINT

1214 D = 1 / T(R,R):T(R,R) = 1:D1
      = 1 / W(R,R):W(R,R) = 1
1216  FOR J = 1 TO M
1218 T(R,J) = T(R,J) * D
1220 W(R,J) = W(R,J) * D1
1222  NEXT J
1224  FOR I = 1 TO M
1226  IF I = R THEN 1238
1228 D = T(I,R):T(I,R) = 0:D1 = W
     (I,R):W(I,R) = 0
```

```
1230  FOR J = 1 TO M
1232 T(I,J) = T(I,J) - D * T(R,J)

1234 W(I,J) = W(I,J) - D1 * W(R,J
     )
1236  NEXT J
1238  NEXT I
1240  IF P3 < 2 THEN 1246
1242 P = 1: GOSUB 1278
1244 P = P3: GOSUB 1384
1246  IF P3 < 3 THEN 1250
1248  GOSUB 1352
1250  GOTO 1152
1252  REM STEP CAL. END
1254  IF L = 0 THEN FAI =  - 1: GOTO
     1276
1256 FAI = 0
1258  IF P3 >  = 4 THEN 1270
1260  IF P1 = 0 THEN 1266
1262  PRINT "L,U,X2="
1264  PRINT L;"  "; FN P(U);"  ";
      FN P(X2)
1266 P = P1: GOSUB 1278
1268 P = P2 + 3: GOSUB 1384
1270  IF P1 < 4 THEN 1276
1272  IF P3 >  = 3 THEN 1276
1274  GOSUB 1352
1276  RETURN
1278  REM SUB.CCD
1280  FOR H = 1 TO G
1282 CO = 0
1284  FOR I = 1 TO M
1286 CI = 0
1288  IF X(I) = 0 THEN 1298
1290  FOR J = 1 TO M
1292  IF X(J) = 0 THEN 1296
1294 CI = CI + W(I,J) * MX(H,J)
1296  NEXT J
```

```
1298 C(H,I) = CI * (N - G)
1300 CO = CO + CI * MX(H,I)
1302  NEXT I
1304 C(H,O) =  - CO * (N - G) / 2

1306  NEXT H
1308  IF P = 0 THEN 4090
1310  PRINT : PRINT : PRINT "C(I,
     J)=": PRINT
1312  FOR I = 1 TO G: PRINT "I=";
     I;":":K = 0: FOR J = 0 TO M:
     Z = C(I,J)
1314  PRINT  FN P(Z),:K = K + 1
1316  IF K = 4 THEN K = 0: PRINT

1318  NEXT J: PRINT : NEXT I: PRINT
     : PRINT
1320 D = 0
1322  FOR H = 2 TO G: FOR K = 1 TO
     H - 1
1324 D = D + 1:D2 = 0
1326  FOR I = 1 TO M
1328  IF X(I) = 0 THEN 1332
1330 D2 = D2 + (C(H,I) - C(K,I)) *
     (MX(H,I) - MX(K,I))
1332  NEXT I
1334 D2HG(D) = D2
1336 FHG(D) = D2 * ((N - G - L +
     1) * NG(H) * NG(K) / (L * (N
      - G) * (NG(H) + NG(K))))
1338  NEXT K: NEXT H
1340  IF P = 0 THEN 1350
1342  PRINT : PRINT "(H,G)"; TAB(
     15);"D2HG="; TAB( 30);"FHG="
     : PRINT
1344 D = 0: FOR H = 2 TO G: FOR K
      = 1 TO H - 1:D = D + 1
```

```
1346  PRINT "(";H;",";K;"):"; TAB(
     15); FN P(D2HG(D)); TAB( 30)
     ; FN P(FHG(D))
1348  NEXT K: NEXT H: PRINT
1350  RETURN
1352  REM SUB. PRINT W,T
1354  PRINT "W(I,J)="
1356  FOR I = 1 TO M: PRINT : PRINT
     "I=";I;":"
1358 K = 0
1360  FOR J = 1 TO M:Z = W(I,J)
1362  PRINT  FN P(Z),
1364 K = K + 1: IF K = 4 THEN K =
     0: PRINT
1366  NEXT J: PRINT : NEXT I: PRINT
     : PRINT
1368  PRINT "T(I,J)="
1370  FOR I = 1 TO M: PRINT : PRINT
     "I=";I;":"
1372 K = 0
1374  FOR J = 1 TO M:Z = T(I,J)
1376  PRINT  FN P(Z),
1378 K = K + 1: IF K = 4 THEN K =
     0: PRINT
1380  NEXT J: PRINT : NEXT I
1382  RETURN
1384  REM SUB.CLASS
1386  FOR I = 1 TO G + 1: FOR J =
     1 TO G + 1:DM(I,J) = 0: NEXT
     J: NEXT I
1388  FOR K = 1 TO N
1390 YMAX =  - 1000000
1392  FOR H = 1 TO G
1394 YH = LQQ(H) + C(H,0)
1396  FOR I = 1 TO M
1398  IF X(I) = 0 THEN 1402
1400 YH = C(H,I) * XY(K,I) + YH
```

```
1402  NEXT I
1404 PR(H)  = YH
1406  IF YH <  = YMAX THEN 141'
1408 YMAX = YH:HMAX = H
1410  NEXT H
1412 D = 0
1414  FOR H = 1 TO G
1416 PR(H) =  EXP (PR(H) - YMAX)
1418 D = D + PR(H)
1420  NEXT H
1422 YMAX = PR(HMAX) / D
1424 H = XY(K,M1)
1426 DM(HMAX,H)  = DM(HMAX,H)  + 1
1428  IF P < 4 THEN 1434
1430  PRINT "K=";K;"H=";H;"HMAX="
     ;HMAX;"YMAX="; FN P(YMAX).
1432  PRINT
1434  NEXT K
1436  FOR H = 1 TO G
1438 D1 = 0:D2 = 0
1440  FOR K = 1 TO G
1442 D1 = D1 + DM(H,K)
1444 D2 = D2 + DM(K,H)
1446  NEXT K
1448 DM(H,G1)  = D1
1450 DM(G1,H)  = D2
1452  NEXT H
1454 DM(G1,G1)  = N
1456  IF P = 0 THEN 1466
1458  PRINT : PRINT "DM(I,J)="
1460  FOR I = 1 TO G + 1: PRINT :
     K = 0: FOR J = 1 TO G + 1:Z =
     DM(I,J): PRINT Z,:K = K + 1
1462  IF K = 9 THEN K = 0: PRINT

1464  NEXT J: PRINT : NEXT I: PRINT

1466  RETURN
```

(5) 例题

① 设有 4 个因子，分成 3 类，各个因子都有17个观测样本，具体数据见下表：

x_1	x_2	x_3	x_4	分类
6	-11.5	19	90	1
-11	-18.5	25	-36	3
90.2	-17	17	3	2
-4	-15	13	54	1
0	-14	20	35	2
0.5	-11.5	19	37	3
-10	-19	21	-42	3
0	-23	5	-35	1
20	-22	8	-20	3
-100	-21.4	7	-15	1
-100	-21.5	15	-40	2
13	-17.2	18	2	2
-5	-18.5	15	18	1
10	-18	14	50	1
-8	-14	16	56	1
0.6	-13	26	21	3
-40	-20	22	-50	3

表中最右面一列分别为对每组观测样本的分类。

设选择 F1=F2=2，要求用逐步判别方法挑选因子构造判别函数。

② 主程序

```
10  DATA 3,4,17,2,2
20  DATA 6,-11,90.2,-4,0,0.5,-10,
```

```
        0,20,-100,-100,13,-5,10,-8,0
        .6,-40
30    DATA -11.5,-18.5,-17,-15,-14,
        -11.5,-19,-23,-22,-21.4,-21.
        5,-17.2
35    DATA -18.5,-18,-14,-13,-20
40    DATA 19,25,17,13,20,19,21,5,8
        ,7,15,18,15,14,16,26,22
50    DATA 90,-36,3,54,35,37,-42,-3
        5,-20,-15,-40,2,18,50,56,21,
        -50
60    DATA 1,3,2,1,2,3,3,1,3,1,2,2,
        1,1,1,3,3
70    GOSUB 1000
80    END
```

③ 计算逐步判别子程序(AP-22)

④ 计算结果

RUN主程序，显示“P1，P2,P3=”,若选择P1=1，P2=0，P3=0，则继续运行，显示“Q(0)=”，若键入0，则又继续运行，打印出如下结果：

```
ĿRUN
P1,P2,P3=1,0,0
Q(0)=0


NG(1)=7
NG(2)=4
NG(3)=6

L,U,X2=
2   .281   17.154


C(I,J)=
```

```
I=1:
-100.389          0              -9.865
0
.953
I=2:
-73.827           0              -8.474
0
.801
I=3:
-61.667           0              -7.74
0
.721


(H,G)             D2HG=          FHG=

(2,1):            4.631          5.473
(3,1):            10.716         16.074
(3,2):            1.259          1.402

DM(I,J)=

7                 0              1
8

0                 4              0
4

0                 0              5
5

7                 4              6
17
```

即计算得出第一类样本数有7个；第二类样本数有4个；第三类样本数有 6 个；最后选中 2 个变量，Wilks＝0.281，$\chi^2=17.154$

第一类判别函数：

$$y_1 \cong -100.39 - 9.86x_2 + 0.95x_4$$

第二类判别函数：

$$y_2 \cong -73.83 - 8.47x_2 + 0.80x_4$$

第三类判别函数：

$$y_3 = -61.67 - 7.74x_2 + 0.72x_4$$

最后打印出判别矩阵，即：

计算＼实况	Ⅰ	Ⅱ	Ⅲ	Σ
Ⅰ	7	0	1	8
Ⅱ	0	4	0	4
Ⅲ	0	0	5	5
Σ	7	4	6	17

可见判别所得拟合较好，只有一次（～6％）错误。

4.1.7 滑动与插值

直线滑动平均

(1) 目的

对于观测序列数值做直线滑动平均，滑动平均所取点可以任意选定。

(2) 数学方法

若试验序列为y_0，y_1，y_2，…，y_i，…，y_m，其中y_i的直线滑动平均值为y'_i

则　$y'_i = a + bt$

式中

$$a=\frac{\sum_{t=-c}^{c} y_{i+t}}{2c+1}$$

$$b=\frac{\sum_{t=-c}^{c} t\,y_{i+t}}{\sum_{t=-c}^{c} t^2}$$

若是取N点滑动平均，则

$$c=INT(N/2)$$

对于$c \leqslant i \leqslant M-c$的序列值，平滑后的值：

$$y_i'=\frac{1}{N}(y_{i-c}+y_{i-c+1}+\cdots+y_i+\cdots+y_{i+c})$$

对于$i<c$的序列值，平滑后为：

$$y_i'=a+bt \qquad (t=-c,\ -c+1,\ \cdots,\ -1)$$

对于$i>M-c$的序列值，平滑后为：

$$y_i'=a+bt \qquad (t=1,\ 2,\ \cdots,\ c)$$

(3) 子程序使用说明

① 简单变量

N——选定做几点滑动平均；

M——序列样本的个数。

② 数组

A(M-1)——原序列各个数值；

B(M-1)——平滑后序列的各个数值。

③ 数据排列顺序

N，M，A(M−1)

④ 计算结果

打印输出线性平滑后序列的各个数值。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB1000转本节子程序。

(4) 子程序

```
1000  REM LIN.SMO.(AP-23)
1002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
1004  READ N,M
1006 M = M - 1
1008  DIM A(M),B(M)
1010  FOR I = 0 TO M
1012  READ A(I)
1014  NEXT I
1016 C =  INT (N / 2)
1018  FOR I = C TO M - C
1020  FOR J =  - C TO C
1022 B(I) = B(I) + A(I + J)
1024  NEXT J
1026 B(I) = B(I) / N
1028  NEXT I
1030  FOR K =  - C TO C
1032 T1 = T1 + K * K
1034 T2 = T2 + A(C + K)
1036 T3 = T3 + K * A(C + K)
1038 T4 = T4 + A(M - C + K)
1040 T5 = T5 + K * A(M - C + K)
1042  NEXT K
1044  FOR I =  - C TO  - 1
1046 B(C + I) = T2 / (2 * C + 1) +
     I * T3 / T1
1048  NEXT I
1050  FOR I = 1 TO C
1052 B(M - C + I) = T4 / (2 * C +
     1) + I * T5 / T1
```

```
1054  NEXT I
1056  FOR I = 0 TO M
1058  PRINT  FN P(B(I)),
1060  NEXT I
1062  PRINT
1064  RETURN
```

(5) 例题

① 若有试验序列，有25个数，计算其7点滑动平均值。

y_i = 1， 2， 1， 2， 3， 2， 4， 3， 5， 4， 3，
6， 7， 6， 6， 8， 9， 11， 9， 10， 11， 10， 12， 14，13

② 主程序

```
10   DATA 7,25
20   DATA 1,2,1,2,3,2,4,3,5,4,3,6,
      7
30   DATA 6,6,8,9,11,9,10,11,10,12
      ,14,13
40   GOSUB 1000
50   END
```

③ 计算滑动平均的子程序(AP-23)

④ 计算结果

```
ÜRUN
.96             1.36            1.75
2.14            2.43            2.86
3.29            3.43            3.86
4.57            4.86            5.29
5.71            6.43            7.57
8               8.43            9.14
9.71            10.29           11
11.29           12.04           12.79
13.54
```

五点三次平滑

(1) 目的

用五点三次平滑公式对等距点上由实验得到的数据进行平滑。

(2) 数学方法

对于n+1个点的平滑方法是：

$$\overline{y_0}=\frac{1}{70}[69y_0+4(y_1+y_3)-6y_2-y_4]$$

$$\overline{y_1}=\frac{1}{35}[2(y_0+y_4)+27y_1+12y_2-8y_3]$$

$$\overline{y_{n-1}}=\frac{1}{35}[2(y_n+y_{n-4})-8y_{n-3}+12y_{n-2}+27y_{n-1}]$$

$$\overline{y_n}=\frac{1}{70}[-y_{n-4}+4(y_{n-3}+y_{n-1})-6y_{n-2}+69y_n]$$

其它点由下式计算：

$$\overline{y_i}=\frac{-3(y_{i-2}+y_{i+2})+12(y_{i-1}+y_{i+1})+17y_i}{35}$$

式中$i=2, 3, \cdots, n-2$

(3) 子程序使用说明

① 简单变量

N——给定数值点的个数，且$N\geqslant 5$。

② 数组

Y(N－1)——开始存放由实验得到的等距点上的数据，最后存放平滑结果。

P(N－1)——子程序中用到的一套工作单元。

③ 数据排列顺序

N，Y(N－1)。

④ 计算结果

打印输出平滑后的数据序列。

⑤ 在主程序的数据语句中，按数据排列顺序给出具体数据后，由GOSUB1000转本节子程序。

(4) 子程序

```
1000  REM PINGHUA53(AP-24)
1002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
1004  READ N
1006 N = N - 1
1008  DIM Y(N),P(N)
1010  FOR I = 0 TO N
1012  READ Y(I)
1014  NEXT I
1016  FOR I = 0 TO N
1018 P(I) = Y(I)
1020  NEXT I
1022 Y(0) = (69 * P(0) + 4 * (P(1
     ) + P(3)) - 6 * P(2) - P(4))
      / 70
1024 Y(1) = (2 * (P(0) + P(4)) +
     27 * P(1) + 12 * P(2) - 8 *
     P(3)) / 35
1026  FOR I = 2 TO N - 2
1028 Y(I) = ( - 3 * (P(I - 2) + P
     (I + 2)) + 12 * (P(I - 1) +
     P(I + 1)) + 17 * P(I)) / 35
1030  NEXT I
1032 Y(N - 1) = (2 * (P(N - 4) +
     P(N)) - 8 * P(N - 3) + 12 *
     P(N - 2) + 27 * P(N - 1)) /
     35
1034 Y(N) = ( - P(N - 4) + 4 * (P
     (N - 3) + P(N - 1)) - 6 * P(
     N - 2) + 69 * P(N)) / 70
```

```
1036  FOR I = 0 TO N
1038  PRINT "Y(";I + 1;")="; FN P
     (Y(I)),
1040  NEXT I
1042  RETURN
```

(5) 例题

① 用五点三次平滑公式平滑下表中的数据：

x_i:	0	1	2	3	4	5	6	7	8
y_i:	54	145	227	359	401	342	259	112	65

② 主程序

```
5   DATA 9
10  DATA 54,145,227,359,401,342,2
     59,112,65
15  GOSUB 1000
20  END
```

③ 五点三次平滑子程序(AP-24)

④ 计算结果

```
ÜRUN
Y(1)=56.84
Y(3)=244.06      Y(2)=133.63
Y(5)=393.46      Y(4)=347.94
Y(7)=241.51      Y(6)=352.03
Y(9)=62.09       Y(8)=123.66
```

牛顿向前、向后插值

(1) 目的

给定函数$y=f(x)$在等距结点$x_k=x_0+kh$上的函数值为$y_k=f(x_k)$, $k=0, 1, \cdots N$；利用牛顿向前或向后插值公式算出被插值点x上的函数值。

(2) 数学方法

利用差分公式算出n阶差分

① 向前差分公式：

$\Delta^n y_k = \Delta^{n-1} y_{k+1} - \Delta^{n-1} y_k$

其中　$n=1,2,\cdots$；　$k=0,1,\cdots,N-1$

② 向后差分公式：

$$\nabla^n y_k = \nabla^{n-1} y_k - \nabla^{n-1} y_{k-1}$$

其中 $n=1,2,\cdots$；　$k=N,N-1,\cdots,1$

牛顿向前插值公式：

令：$t=(x-x_0)/h$

即　$x=x_0+th$；则有

$$f(x)=f(x_0+th)=y_0+\frac{t}{1!}\Delta y_0+\frac{t(t+1)}{2!}\Delta^2 y_0+$$

$$\cdots\frac{t(t-1)(t-2)\cdots(t-n+1)}{n!}\Delta^n y_0$$

牛顿向后插值公式：

令　$x=x_N+th$，则有

$$f(x)=f(x_N+th)=y_N+\frac{t}{1!}\nabla y_N+\frac{t(t+1)}{2!}\nabla^2 y_N$$

$$+\cdots+\frac{t(t+1)(t+2)\cdots(t+n-1)}{n!}\nabla^n y_N$$

$$=y_N+\frac{t}{1!}\Delta y_{N-1}+\frac{t(t+1)}{2!}\Delta^2 y_{N-2}+\cdots$$

$$+\frac{t(t+1)(t+2)\cdots(t+n-1)}{n!}\Delta^n y_{N-n}$$

(3) 子程序使用说明

根据差分的特性可知，当差分阶数达到某个值时，例如 $r+1$ 阶差分，则有：$|\Delta^{r+1} y_k| < 2^r \times 10^{-m}$，其中 m 为 y_k 所具有的小数位数；而所有小于等于 r 阶的差分都不满足

上述条件。向后差分也有类似的特性。我们在使用差分时，只应使用到 r 阶差分。为此，在本节子程序中引用了控制参数 $E \leqslant 2^{r} * 10^{-m}$。

当被插值点 x 给定后，本子程序选取靠近 x 的五个结点进行插值，当 x 给在区间 $[x_i, x_{i+1}]$ 上，$i + r (= P = 4) \leqslant N$ 时，则使用牛顿向前插值公式；当 $i + r > N$ 时，则使用牛顿向后插值公式。

① 简单变量

M——给定的被插值点的个数减一；

N——给定的等距结点的个数减一；

H——给定的等距结点间的长度；

E——控制差分的阶数，使 r + 1 阶差分的多数 $\left(\frac{8}{10}\right)$ 差分值满足 $|\Delta^{r+1} y_k| \leqslant E$；

P——差分的阶数（本子程序最高为四阶）；

N 2——初值为给定的等距结点的个数，为程序中计算某阶差分的差分值个数。

② 数组

X(N)——存放给定的结点值；

Y(N)——存放给定结点上的函数值；

Z(M)——存放给定的被插值点的值；

A(N)——存放一阶差分值；

B(N)——存放二阶差分值；

C(N)——存放三阶差分值；

D(N)——存放四阶差分值；

T(N)——计算差分子程序中，存放各阶差分值；

Q(N)——计算差分子程序中，存放前一阶差分值；

F(M)——存放被插值点的函数值；

R(P)——存放插值多项式的各阶差分值。

③ 数据排列顺序

M，N，H，E，P，N2，X(N)，Y(N)，Z(M)。

④ 计算结果

打印输出被插值点的各个函数值。

⑤ 在主程序的数据语句中，按数据排列顺序给出数据，然后由GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM NEWTON(AP-25)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ M,N,H,E,P,N2
1006  DIM X(N),Y(N),Z(M),A(N),B(N
     ),C(N),D(N),T(N),Q(N),F(M),R
     (P)
1008  FOR I = 0 TO N
1010  READ X(I)
1012  NEXT I
1014  FOR I = 0 TO N
1016  READ Y(I)
1018  NEXT I
1020  FOR I = 0 TO M
1022  READ Z(I)
1024  NEXT I
1026  FOR I = 0 TO N
1028 Q(I) = Y(I)
1030  NEXT I
1032  GOSUB 1102
1034  FOR I = 0 TO N
1036 A(I) = T(I)
1038  NEXT I
1040 K1 =  INT (N2 * 0.9)
1042  IF NO > K1 THEN 1122
```

```
1044  FOR I = 0 TO N
1046 Q(I) = A(I)
1048  NEXT I
1050  GOSUB 1102
1052  FOR I = 0 TO N
1054 B(I) = T(I)
1056  NEXT I
1058 K1 =  INT (N2 * 0.9)
1060  IF NO > K1 THEN 1122
1062  FOR I = 0 TO N
1064 Q(I) = B(I)
1066  NEXT I
1068  GOSUB 1102
1070  FOR I = 0 TO N
1072 C(I) = T(I)
1074  NEXT I
1076 K1 =  INT (N2 * 0.9)
1078  IF NO > K1 THEN 1122
1080  FOR I = 0 TO N
1082 Q(I) = C(I)
1084  NEXT I
1086  GOSUB 1102
1088  FOR I = 0 TO N
1090 D(I) = T(I)
1092  NEXT I
1094 K1 =  INT (N2 * 0.9)
1096  IF NO > K1 THEN 1122
1098  PRINT "UJ"
1100  GOTO 1122
1102 N1 = N1 + 1
1104 N2 = N2 - 1
1106 NO = 0
1108  FOR I = 0 TO N2
1110 T(I) = Q(I + 1) - Q(I)
1112 X1 = T(I)
1114  IF  ABS (T(I)) > E THEN 111
     8
1116 NO = NO + 1
```

```
1118  NEXT I
1120  RETURN
1122  FOR J = 0 TO M
1124  FOR I = 0 TO N - 1
1126  IF X(I) > Z(J) THEN 1138
1128  IF X(I + 1) < Z(J) THEN 113
     6
1130 S = I
1132 R(1) = A(S)
1134  GOTO 1146
1136  NEXT I
1138  PRINT "BUZI"
1140  STOP
1142  INPUT Z(J)
1144  GOTO 1124
1146  IF S + N1 > N THEN 1156
1148 R(2) = B(S)
1150 R(3) = C(S)
1152 R(4) = D(S)
1154  GOTO 1162
1156 R(2) = B(S - 1)
1158 R(3) = C(S - 2)
1160 R(4) = D(S - 3)
1162 K = 1
1164 H1 = 1
1166  IF S + N1 > N THEN 1174
1168 F(J) = Y(S)
1170 T1 = (Z(J) - X(S)) / H
1172  GOTO 1178
1174 F(J) = Y(S + 1)
1176 T1 = (Z(J) - X(S + 1)) / H
1178  FOR I = 1 TO N1
1180  IF S + N1 > N THEN 1186
1182 K = K * (T1 - I + 1)
1184  GOTO 1188
1186 K = K * (T1 + I - 1)
1188 H1 = H1 * I
1190 F(J) = F(J) + (K / H1) * R(I
```

```
        )
1192  NEXT I
1194  NEXT J
1196  FOR I = 0 TO M
1198  PRINT  FN P(F(I))
1200  NEXT I
1202  PRINT
1204  RETURN
```

(5) 例题

① 给定等距结点 $x_k = 4000 + K \times 500$ 及对应的函数值 y_k（y_k的具体数值见主程序中数据语句：15 DATA），k＝0，1，… 14。求被插值点z_j（具体数值见主程序）上的函数值：$f(z_j)=?$ $j=0,1,2$。

② 主程序

```
5.  DATA 2,14,500,0.1,4,14
10   DATA 4000,4500,5000,5500,6000
       ,6500,7000,7500,8000
13   DATA 8500,9000,9500,10000,105
       00,11000
15   DATA 1.38,1.48,1.58,1.69,1.81
       ,1.94,2.10,2.28,2.50
17   DATA 2.76,3.06,3.41,3.83,4.33
       ,4.93
20   DATA 5200,7200,10700
25   GOSUB 1000
30   END
```

③ 牛顿插值方法子程序（AP-25）

④ 计算结果

1.623　　2.167　　4.558

拉格朗日插值

(1) 目的

设已知函数y(x)的结点值为x_i（$i=0, 1 \cdots, n$）及对

应的函数值$y_i(i=0,1,\cdots,n)$，对于给定的不是结点的x，用一元拉格朗日n+1点插值公式计算对应的函数值y(x)。

(2) 数学方法

插值公式是：

$$y(x)=\sum_{j=0}^{n}\prod_{\substack{i=0\\ i\neq j}}^{n}\left(\frac{x-x_i}{x_j-x_i}\right)y_j$$

本节子程序是对m+1个不是结点的值$x_k(k=0,1,\cdots,m)$进行成组插值，求出其对应的函数值$y_k=y(x_k)$

(3) 子程序使用说明

① 简单变量

N——给定的插值结点个数；

M——要插值的函数个数。

② 数组

X(N-1)——存放给定的插值结点值；

Y(N-1)——存放给定插值结点上的函数值；

U(M-1)——存放被插值的结点值；

V(M-1)——存放插值结果。

③ 数据排列顺序

N, M, X(N-1), Y(N-1), U(M-1)

④ 计算结果

打印输出插值结果。

⑤ 在主程序中按数据排列顺序给出数据后，由GOSUB 1000转本节子程序

(4) 子程序

```
1000  REM LAGRANGE(AP-26)
1002  DEF  FN P(Y) =  INT (Y * 10
```

```
      00 + 0.5) / 1000
1004  READ N,M
1006 N = N - 1
1008 M = M - 1
1010  DIM X(N),Y(N),U(M),V(M)
1012  FOR I = 0 TO N
1014  READ X(I)
1016  NEXT I
1018  FOR I = 0 TO N
1020  READ Y(I)
1022  NEXT I
1024  FOR I = 0 TO M
1026  READ U(I)
1028  NEXT I
1030  FOR K = 0 TO M
1032  FOR J = 0 TO N
1034 L = 1
1036  FOR I = 0 TO N
1038  IF I = J THEN 1042
1040 L = L * (U(K) - X(I)) / (X(J
     ) - X(I))
1042  NEXT I
1044 V(K) = V(K) + L * Y(J)
1046  NEXT J
1048  NEXT K
1050  FOR I = 0 TO M
1052  PRINT  FN F(V(I)),
1054  NEXT I
1056  PRINT
1058  RETURN
```

(5) 例题

① 已知函数如下表：

x： $-2, -1, 0, 1, 2$

y： $-4, -2, 0, 2, 4$

求当$x = -1.5, -0.5, 0.5, 1.5$时对应的函数值。

② 主程序

```
5   DATA 5,4
10   DATA -2,-1,0,1,2,-4,-2,0,2,4
20   DATA -1.5,-0.5,0.5,1.5
30   GOSUB 1000
40   END
```

③ 拉格朗日插值子程序（AP-26）

④ 计算结果

－3　　－1　　1

　3

一元三点插值

(1) 目的

已知函数$y(x)$的结点值为$x_i(i=0,1,\cdots,n)$其对应的函数值为$y(x_i)=y_i$ $(i=0,1,\cdots,n)$，对于给定的不是结点的值 x，选取最靠近它的三个插值点，应用一元拉格朗日三点插值公式计算出对应的函数值$y(x)$。

(2) 数学方法

插值公式：

$$y(x)=\sum_{j=k}^{k+2}\prod_{\substack{i=k\\i\neq j}}^{k+2}\left(\frac{x-x_i}{x_j-x_i}\right)y_j$$

式中：

$k=0$　当$x\leqslant x_1$时

$\quad s$　当$x_s<x\leqslant x_{s+1}$，$x-x_s\geqslant x_{s+1}-x$时

$\quad\quad(s=1,2,\cdots,n-2)$

$\quad s-1$　当$x_s<x\leqslant x_{s+1}$，$x-x_s<x_{s+1}-x$时

$\quad\quad(s=1,2,\cdots,n-2)$

$\quad n-2$　当$x\geqslant x_{n-1}$时

本子程序是对m＋1个不是结点的值x_l（$l=0, 1, \cdots, m$）进行成组插值，求出对应的函数值$y_l = y(x_l)$。

（3）子程序使用说明

① 简单变量

N——给定的插值结点的个数；

M——被插值点的个数。

② 数组

X(N−1)——存放给定的插值结点值；

Y(N−1)——存放给定的插值结点上的函数值；

U(M−1)——存放被插值点的值；

V(M−1)——存放插值结果。

③ 数据排列顺序

N，M，X(N−1)，Y(N−1)，U(M−1)

④ 计算结果

打印输出插值结果。

⑤ 在主程序的数据语句中，按数据排列顺序给出具体数值后，由GOSUB 1000转本节子程序。

（4）子程序

```
1000  REM LAGRANGE13(AP-27)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ N,M
1006 N = N - 1
1008 M = M - 1
1010  DIM X(N),Y(N),U(M),V(M)
1012  FOR I = 0 TO N
1014  READ X(I)
1016  NEXT I
1018  FOR I = 0 TO N
1020  READ Y(I)
```

```
1022  NEXT I
1024  FOR I = 0 TO M
1026  READ U(I)
1028  NEXT I
1030  FOR L = 0 TO M
1032  FOR K = 0 TO N - 3
1034  IF U(L) <  = X(K + 1) THEN
     1040
1036  NEXT K
1038 K = N - 2
1040  IF K = 0 THEN 1046
1042  IF U(L) - X(K) >  = X(K + 1
     ) - U(L) THEN 1046
1044 K = K - 1
1046 X0 = X(K)
1048 X1 = X(K + 1)
1050 X2 = X(K + 2)
1052 P = (U(L) - X1) * (U(L) - X2
     ) / ((X0 - X1) * (X0 - X2))
1054 Q = (U(L) - X0) * (U(L) - X2
     ) / ((X1 - X0) * (X1 - X2))
1056 R = (U(L) - X0) * (U(L) - X1
     ) / ((X2 - X0) * (X2 - X1))
1058 V(L) = P * Y(K) + Q * Y(K +
     1) + R * Y(K + 2)
1060  NEXT L
1062  FOR I = 0 TO M
1064  PRINT "Y(";U(I);")="; FN P(
     V(I)),
1066  NEXT I
1068  RETURN
```

（5）例题

① 已知函数如下表：

x： 0.2， 0.24， 0.28， 0.32， 0.36， 0.40

y： 0.19867，0.2377，0.27636，0.31457，0.35227，0.38942

求：当 $x=0.22$，0.29，0.38时的 y 值。

② 主程序

```
5  DATA 6,3
10  DATA 0.2,0.24,0.28,0.32,0.36,
     0.40
15  DATA 0.19867,0.2377,0.27636,0
     .31457,0.35227,0.38942
20  DATA 0.22,0.29,0.38
25  GOSUB 1000
30  END
```

③ 一元三点插值子程序(AP-27)

④ 计算结果

```
]RUN
Y(.22)=.218        Y(.29)=.286
Y(.38)=.371
```

二元三点插值

(1) 目的

已知函数 $f(x, y)$ 的第一变量(x)的结点值为 x_i(不一定等距，$i=0, 1, \cdots, n$)，第二变量(y)的结点值为 y_j(不一定等距，$j=0, 1, \cdots, m$)，其对应结点上的函数值为 $f_{ij}(i=0, 1, \cdots, n;\ j=0, 1, \cdots, m)$；对于给定的不是结点的值 (x, y).分别选取最靠近 x 的三个点(x_q, x_{q+1}, x_{q+2})和最靠近 y 的三个点(y_p, y_{p+1}, y_{p+2})，用二元拉格朗日插值公式计算出对应的函数值 $f(x, y)$。

(2) 数学方法

插值公式：

$$f(x, y)=\sum_{i=q}^{q+2}\sum_{j=p}^{p+2}\left(\prod_{\substack{S=q\\ S\neq i}}^{q+2}\frac{x-x_s}{x_i-x_s}\right)$$

$$\cdot \left(\prod_{\substack{l=p \\ l \neq j}}^{p+2} \frac{y - y_l}{y_j - y_l} \right) f_{ij}$$

式中 i， j 选取方法与一元三点插值相同。本节介绍的方法可对 l + 1 个不是结点的变元值 (x_k, y_k) （ k = 0， 1， …， l ）进行成组插值，求出对应的函数值。

(3) 子程序使用说明

① 简单变量

N —— x 方向上给定的插值结点个数；

M —— y 方向上给定的插值结点个数；

L ——被插值点的个数。

② 数组

X(N-1)——存放 x 方向上给定的插值结点值；

Y(M-1)——存放 y 方向上给定的插值结点值；

F(N-1, M-1)——存放给定插值结点上的函数值；

S(L-1)——存放 x 方向上要插值的点的值；

T(L-1)——存放 y 方向上要插值的点的值；

W(L-1)——存放插值结果。

③ 数据排列顺序

N， M， L， X(N-1)， Y(M-1)，

F(N-1, M-1)， S(L-1)， T(L-1)

④ 计算结果

打印输出插值结果。

⑤ 在主程序中按数据排列顺序给出数据后，由 GOSUB 1000转本节子程序。

(4) 子程序

```
1000  REM LAGRANGE23(AP-28)
1002  DEF  FN P(Y) = INT (Y * 10
  .   00 + 0.5) / 1000
1004  READ N,M,L
1006 N = N - 1
1008 M = M - 1
1010 L = L - 1
1012  DIM X(N),Y(M),F(N,M),S(L),T
      (L),W(L),U(2),V(2)
1014  FOR I = 0 TO N
1016  READ X(I)
1018  NEXT I
1020  FOR I = 0 TO M
1022  READ Y(I)
1024  NEXT I
1026  FOR I = 0 TO N
1028  FOR J = 0 TO M
1030  READ F(I,J)
1032  NEXT J
1034  NEXT I
1036  FOR I = 0 TO L
1038  READ S(I)
1040  NEXT I
1042  FOR I = 0 TO L
1044  READ T(I)
1046  NEXT I
1048  FOR K = 0 TO L
1050  FOR I = 0 TO N - 3
1052  IF S(K) <  = X(I + 1)  THEN
      1058
1054  NEXT I
1056 I = N - 2
1058  FOR J = 0 TO M - 3
1060  IF T(K) <  = Y(J + 1)  THEN
      1066
1062  NEXT J
1064 J = M - 2
1066  IF I = 0 THEN 1072
```

```
1068  IF S(K) - X(I) >  = X(I + 1
     ) - S(K) THEN 1072
1070 I = I - 1
1072  IF J = 0 THEN 1078
1074  IF T(K) - Y(J) >  = Y(J + 1
     ) - T(K) THEN 1078
1076 J = J - 1
1078 X0 = X(I)
1080 X1 = X(I + 1)
1082 X2 = X(I + 2)
1084 Y0 = Y(J)
1086 Y1 = Y(J + 1)
1088 Y2 = Y(J + 2)
1090 U(0) = (S(K) - X1) * (S(K) -
     X2) / ((X0 - X1) * (X0 - X2)
     )
1092 U(1) = (S(K) - X0) * (S(K) -
     X2) / ((X1 - X0) * (X1 - X2)
     )
1094 U(2) = (S(K) - X0) * (S(K) -
     X1) / ((X2 - X0) * (X2 - X1)
     )
1096 V(0) = (T(K) - Y1) * (T(K) -
     Y2) / ((Y0 - Y1) * (Y0 - Y2)
     )
1098 V(1) = (T(K) - Y0) * (T(K) -
     Y2) / ((Y1 - Y0) * (Y1 - Y2)
     )
1100 V(2) = (T(K) - Y0) * (T(K) -
     Y1) / ((Y2 - Y0) * (Y2 - Y1)
     )
1102  FOR Q = 0 TO 2
1104  FOR P = 0 TO 2
1106 W(K) = W(K) + U(Q) * V(P) *
     F(I + Q,J + P)
1108  NEXT P
1110  NEXT Q
1112  NEXT K
```

```
1114  FOR I = 0 TO L
1116  PRINT "f(";S(I);",";T(I);")
      ="; FN P(W(I)),
1118  NEXT I
1120  RETURN
```

(5) 例题

① 已知函数如下表所示：

x_i \ f_{ij} \ y_j	25	35	45
10	0.43674	0.61193	0.78756
30	0.43973	0.62003	0.80437
50	0.44455	0.63364	0.83431
70	0.44904	0.64707	0.86653

求当 x, y 分别为(40, 30), (20, 30), (60, 40)时的函数值 f(x, y)。

② 主程序

```
5   DATA 4,3,3
10  DATA 10,30,50,70,25,35,45
15  DATA 0.43674,0.61193,0.78756,
      0.43973,0.62003,0.80437,0.44
      455,0.63364,0.83431
20  DATA 0.44904,0.64707,0.86653
25  DATA 40,20,60,30,30,40
30  GOSUB 1000
35  END
```

③ 二元三点插值子程序(AP-28)

④ 计算结果

```
ÜRUN
f(40,30)=.534     f(20,30)=.526
f(60,40)=.743
```

4.1.8 曲线拟合

用正交多项式拟合曲线

(1) 目的

对于等距结点 $x_i(i=0, 1, \cdots, n)$ 上的函数值 $y_i(i=0, 1, \cdots, n)$，用正交多项式以一定的精度拟合。

(2) 数学方法

已知等距结点 x_i $(i=0, 1, \cdots, n)$ 上的函数值为 $y_i(i=0, 1, \cdots, n)$，$x_0=x_1$，结点间距是H。可以用如下多项式拟合函数 y：

$$y=b_0\psi_0(x)+b_1\psi_1(x)+$$
$$+b_2\psi_2(x)+\cdots+b_m\psi_m(x)$$

在本节子程序中，我们最多只用三次多项式拟合实验数据，即令：

$$y=b_0\psi_0(x)+b_1\psi_1(x)+b_2\psi_2(x)+b_3\psi_3(x)$$

构成一个正交多项式：

$$\psi_0(t)=1$$
$$\psi_1(t)=t$$
$$\psi_{k+1}(t)=\psi_1(t)\psi_k(t)-$$
$$-\frac{k^2-(n^2-k^2)}{4(4k^2-1)}\psi_{k-1}(t)$$

系数

$$b_i=\frac{\Sigma\psi_i(I)Y_I}{S_i}=\frac{B_i}{S_i}$$

$(i=0, 1, 2, 3)$

其中 $S_i = \sum \psi_i^2$

$$M_i = b_i B_i$$

$$MS = \sum_{i=0}^{3} M_i$$

$$YS = \sum_{i=0}^{N} Y_i^2$$

误差D是：

$$D = |YS - MS|$$

统计量 F_i 是：

$$F_i = \frac{M_i}{\frac{D}{N-4}} \quad (i=0, 1, 2, 3)$$

第一自由度是1，第二自由度是 $N-4$

若 $F_i > F_{0.01}$ 则该项系数显著。

最后通过变换可以得到函数表达式是：

$$y = A_0 + A_1 x + A_2 x^2 + A_3 x^3$$

(3) 子程序使用说明

① 简单变量

N——给定的结点的个数；

X1——x序列第一个数值；

H——x序列的间距。

② 数组

Y(N)——y序列的各个观测数据；

F(3)——对多项式中四个系数分别得出的统计量。

③ 数据排列顺序

N, X1， H， Y(N)。

④ 计算结果

打印输出拟合的三次多项式中检验四个系数显著性的统计量F0， F1， F2， F3（若误差＜10^{-10}则不打印这几个统计量），最后打印出多项表达式。

⑤ 在主程序的数据语句中，按数据排列顺序给出具体数据后，由GOSUB1000转本节子程序。

(4) 子程序

```
1000  REM CUR.ANA.(AP-29)
1002  DEF  FN P(Y) =  INT (Y * 10
     0 + 0.5) / 100
1004  READ N,X1,H
1006  DIM Y(N),A(3,N),S(3)
1008  DIM B(3),BA(3),M(3),F(3)
1010  FOR I = 1 TO N
1012  READ Y(I)
1014  NEXT I
1016  FOR J = 1 TO N
1018 A(0,J) = 1
1020 A(1,J) = J - (N + 1) / 2
1022  NEXT J
1024  FOR I = 2 TO 3
1026  FOR J = 1 TO N
1028 A(I,J) = A(I - 1,J) * A(1,J)
      - (I - 1) ^ 2 * (N * N - (I
      - 1) ^ 2) * A(I - 2,J) / (4
      * (4 * (I - 1) ^ 2 - 1))
1030  NEXT J
1032  NEXT I
1034  FOR I = 0 TO 3
1036  FOR J = 1 TO N
1038 S(I) = S(I) + A(I,J) * A(I,J
     )
1040 B(I) = B(I) + A(I,J) * Y(J)
1042  NEXT J
```

```
1044 BA(I) = B(I) / S(I)
1046 M(I) = BA(I) * B(I)
1048 MS = MS + M(I)
1050  NEXT I
1052  FOR I = 1 TO N
1054 YS = YS + Y(I) * Y(I)
1056  NEXT I
1058 D =  ABS (YS - MS)
1060  IF D < 10E - 10 THEN 1072
1062  FOR I = 0 TO 3
1064 F(I) = M(I) * (N - 4) / D
1066  PRINT "F";I;"="; FN P(F(I))
     ;
1068  NEXT I
1070  PRINT
1072 A1 = BA(1) - BA(2) * (N + 1)
      + BA(3) * (0.6 * N * N + 1.
     5 * N + 1.1)
1074 A2 = BA(2) - BA(3) * 3 * (N +
     1) / 2
1076 A3 = BA(3)
1078 A0 = BA(0) - BA(1) * (N + 1)
      * 0.5 + BA(2) * (N ^ 2 + 3 *
     N + 2) / 6 - BA(3) * (N + 1)
      * (N + 2) * (N + 3) * 0.05
1080 C = 1 - X1 / H
1082 B0 = A0 + A1 * C + A2 * C *
     C + A3 * C * C * C
1084 B1 = (A1 + 2 * A2 * C + 3 *
     A3 * C * C) / H
1086 B2 = (A2 + 3 * A3 * C) / (H *
     H)
1088 B3 = A3 / (H * H * H)
1090  PRINT "Y="; FN P(B0);"+("; FN
     P(B1);")*X+(";
1092  PRINT  FN P(B2);")*X^2+("; FN
     P(B3);")*X^3"
1094  RETURN
```

（5）例题

① 设N＝7， x_0＝1， H＝1

x_i:	1	2	3	4	5	6	7
y_i:	4	2	1	2	3	6	10

求函数y的表达式。

② 主程序

```
10   DATA 7,1,1
20   DATA 4,2,1,2,3,6,10
30   GOSUB 1000
40   END
```

③ 曲线拟合子程序（AP－29）

④ 计算结果

```
]RUN
F0=1411.2              F1=352.8
F2=375                 F3=0
Y=7.14+(-3.76)*X+(.6)*X^2+(0)*X^3
```

第一自由度是1，第二自由度是3，查F表，$F_{0.01}(1, 3)=34$，即F0，F1及F2都大于 $F_{0.01}$ 故可用二次正交多项式拟合函数 $y=7.14-3.76x+0.60x^2$。

4.1.9 过程分析

周期分析

（1）目的

对一个时间序列，提取其主要周期。

（2）数学方法

用三角函数法提取周期。设有一个时间序列x(t)：x_1，x_2，…，x_n

可以用三角函数族提取周期：

$$x(t)=a_0+\sum_{j=1}^{k}\left(a_j\cos\frac{2\pi t}{L_j}+b_j\sin\frac{2\pi t}{L_j}\right)$$

其中a_0，a_j，b_j为待定系数，L_j为周期的波长。j 为波数，$j=1, 2, 3, \cdots, k$

L_j为周期，$L_j=\frac{n}{1}, \frac{n}{2}, \frac{n}{3}, \cdots, \frac{n}{k}$

$k=INT(n/2)$

$$a_0=\frac{1}{n}\sum_{i=1}^{n}x_i$$

$$a_j=\frac{2}{n}\sum_{i=1}^{n}x_i\cos\left[\frac{2\pi(i-1)}{n}j\right]$$

$$b_j=\frac{2}{n}\sum_{i=1}^{n}x_i\sin\left[\frac{2\pi(i-1)}{n}j\right]$$

振幅$A_j^2=a_j^2+b_j^2$

对这些周期可用Fisher判据来判断：

令 $S_j^2=\frac{1}{2}A_j^2$

S^2 为时间序列x(t)的方差：

$$S^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})^2=$$

$$\frac{1}{n}\sum_{i=1}^{n} x_j^2-\frac{1}{n^2}\left(\sum_{i=1}^{n} x_i\right)^2$$

计算统计量$y_j=S_j^2/S^2$，并将y_j按大小顺序排列，按设计提取10个主波，顺次计算概率P_j，Fisher分布概率公式为：

$$P_j\ (y-y_j)=\sum_{i=1}^{r}(-1)^{i+1}$$

$$\cdot C_{k-j+1}^{i+1}\ [1-(i+1)Y_j]^{k-j}$$

其中$r=INT\ ((1-Y_j)/Y_j)$

若$P_j<0.05$，则主要周期存在；$P_j\geq 0.05$时主要周期不存在。

(3)、子程序使用说明

① 简单变量

N——序列数据的个数。

② 数组

X (N) —— 序列的各个观测数据；

A1(K),B1(K) ——周期函数中的待定系数；

AJ(K) ——振幅；

WN(K) ——波数；

Y(K) ——统计量；

R(10) ——计算Y(K) 时用到的参数；

P1(10) ——概率值

③ 数据排列顺序

N，X(N)

④ 计算结果

打印输出：

A 0——周期函数的系数；

P——各主要周期的概率值；

WAVE NUM.——各主要周期的波数；

PERIOD——各主要周期波长；

A，B——各主要周期函数的系数。

⑤ 在主程序中按数据排列顺序给出数据后，由GOS－UB1000转本节子程序。

（4）子程序

```
1000  REM PER.ANA.(AF-30)
1002  DEF  FN P(Y) =  INT (Y * 10
     00 + 0.5) / 1000
1004  READ N
1006  DIM X(N)
1008  FOR I = 1 TO N
1010  READ X(I)
1012  NEXT I
1014 K =  INT (N / 2)
1016  DIM A1(K),B1(K),AJ(K),WN(K)
1018  DIM Y(K),R(10),P1(10)
1020  FOR I = 1 TO N
1022 S1 = S1 + X(I)
1024 S2 = S2 + X(I) ^ 2
1026  NEXT I
1028 A0 = S1 / N
1030 SS = (S2 - S1 ^ 2 / N) / N
1032  FOR J = 1 TO K
1034  FOR I = 1 TO N
1036 A1(J) = A1(J) + X(I) *  COS
     (2 * 3.14159 * (I - 1) * J /
     N)
1038 B1(J) = B1(J) + X(I) *  SIN
     (2 * 3.14159 * (I - 1) * J /
     N)
1040  NEXT I
```

```
1042 A1(J) = 2 * A1(J) / N
1044 B1(J) = 2 * B1(J) / N
1046 AJ(J) = A1(J) ^ 2 + B1(J) ^
     2
1048 WN(J) = J
1050 Y(J) = AJ(J) / (2 * SS)
1052  NEXT J
1054  FOR I = 1 TO K
1056  FOR J = I TO K
1058  IF Y(I) > = Y(J) THEN 1090

1060 T1 = Y(I)
1062 Y(I) = Y(J)
1064 Y(J) = T1
1066 T1 = AJ(I)
1068 AJ(I) = AJ(J)
1070 AJ(J) = T1
1072 T1 = WN(I)
1074 WN(I) = WN(J)
1076 WN(J) = T1
1078 T1 = A1(I)
1080 A1(I) = A1(J)
1082 A1(J) = T1
1084 T1 = B1(I)
1086 B1(I) = B1(J)
1088 B1(J) = T1
1090  NEXT J
1092  NEXT I
1094  PRINT "AO="; FN P(AO)
1096  FOR I = 1 TO 10
1098 CC = (K - I + 1) * (K - I) /
     2
1100 R(I) =  INT ((1 - Y(I)) / Y(
     I))
1102  FOR J = 1 TO R(I)
1104 P1(I) = P1(I) + ( - 1) ^ (J +
     1) * (1 - (J + 1) * Y(I)) ^
     (K - I) * CC
1106 CC = CC * (K - I - J) / (J +
```

```
        2)
1108  NEXT J
1110  IF P1(I) > = 0.05 THEN PRINT
      "P(";I;")="; FN P(P1(I)); SPC(
      10);"NO PERIOD": GOTO 1122
1112  PRINT "P(";I;")="; FN P(P1(
      I))
1114  PRINT "WAVE NUM.="; FN P(WN
      (I)); SPC( 10);"PERIOD="; FN
      P(N / WN(I))


1116  PRINT "A(";I;")="; FN P(A1(
      I));"*COS("; FN P(360 * WN(I
      ) / N);"*t)"; SPC( 10);
1118  PRINT "B(";I;")="; FN P(B1(
      I));"*SIN("; FN P(360 * WN(I
      ) / N);"*t)"
1120  NEXT I
1122  RETURN
```

（5）例题

① 设有一个时间序列：126，89，115，79，131，89，98，75，109，80，99，99，62，86，97，109，123，122，128，101，求周期。

② 主程序

```
10   DATA 20
20   DATA 126,89,115,79,131,89,98,
      75,109,80
30   DATA 99,99,62,86,97,109,123,1
      22,128,101
40   GOSUB 1000
50   END
```

③ 周期分析子程序（AP-30）

④ 计算结果

```
AO=100.85
P(1)=1E-03
WAVE NUM.=10            PERIOD=2
A(1)=15.9*COS(180*t)            B(1)=0*SIN(180*t)
P(2)=.033
WAVE NUM.=1             PERIOD=20
A(2)=13.745*COS(18*t)           B(2)=-4.472*SIN(18*t)
P(3)=.543               NO PERIOD
```

由于P（1）＜0.05，故存在第一主要周期，波数为10，波长为2；

由于P（2）＜0.05，故存在第二主要周期，波数为1，波长为20；

由于P（3）＞0.05，故不存在第三周期。

观测数据可由主要周期的叠加得到，即：

$$x(t)=A_0+A(1)+B(1)+A(2)+B(2)$$
$$=100.85+15.9\cos(180t)+13.745\cos(18t)-4.472\sin(18t)$$

式中三角函数中的角度以度为单位，t代表序列编号（t＝1,2,…）。

§4.2 人事管理程序

4.2.1 程序设计思路

某些单位的人事部门想要记录所有职工的情况，并希望能够随时更新这些记录。比方说，增加新职工情况、删除调走的人员情况、更改职工有关项目，还希望很方便的针对某些项目对人员情况进行调查、选择、归类。最后，还希望能够随时把这些记录存在磁盘上或从磁盘上取出。针对上述这些要求，我们编制了人事管理软件。

4.2.2 建立汉字资料库

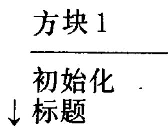
在APPLE Ⅱ微型计算机上配汉字插件以及相应的打印机，即可进行一般性的中文信息处理。为了使用方便和易于推广，本程序是在汉字插件的基础上编制而成的。为使计算机能够处理中文信息，首先要解决汉字进入计算机即转换成机器代码问题。汉字随着历史发展不断演变，且古今繁简，正异各体并存，大约有六万多个汉字，而我们常用的汉字只是汉字集合中的一个子集。APPLE Ⅱ微型机上应用的汉字插件大致可产生24000左右个组合汉字。据说是依《康熙字典》而汇成，故其汉字为繁体。最近，市场上也已推出简体汉字插件。

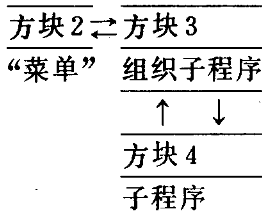
中文输入是利用键盘上英文字母键A—Y代表中文字母来组成汉字，由CTRL-L键来决定输入英文字母还是中文字母。按下CTRL-D键，再按CTRL-L时，屏幕左下角会显现出〈中文〉两个小字，此时A—Y键代表中文字母日—卜，在每输入一中文字母时，均会在〈中文〉的右边依次显示出，每个中文字最多由五个中文字母组成。输入完字母，按空格键，该字即显现出。中文输入完毕时，再按一次CTRL-L键恢复英文输入方式。

汉字的编码组成规律以及详细情况，请参考《佳佳汉卡中文说明》、《汉字库使用手册》。本软件建立汉字资料库采用专门程序（在盘上提供）进行，在后面详述。

4.2.3 程序结构

程序结构有四个主要方块，如下图所示：





我们先简单的介绍每一个方块，然后再详细说明每一个子程序。

方块1是程序的初始部分，本程序采用自动启动输入方式。另外，在这部分向使用者介绍了该程序的标题及其有关情况。在编制程序时，这部分往往容易被忽略掉，这是不应该的，因为虽然没有它并不会影响到程序的逻辑运行，但是却会使程序更难使用、追踪和了解。况且应用汉字插件编制程序时，某些初始化工作更是必不可少的。当然，结合所编程序的不同，这部分具备的要素也会有所不同。

方块2是一个包含程序所有功能的表，常常被称为"菜单"(menu)。使用者可利用这个表来选择所希望具有的功能。它是程序的中心部分，所有的路径都从它开始，而且它们最后还回到这里。因此，如果使用者从这个"菜单"选择了某一项功能，程序会去做这项工作，当这项工作完成之后，它又回到这"菜单"上来。

方块3是为完成某项具体工作而组织子程序。当使用者从"菜单"中选择了一项功能后，为成功的完成该项工作，程序常常必须调用一些子程序。方块3正是为了"菜单"中的每一道"菜"分别组织这个过程。当然，在任何这种组织里，最后一条指令都是把程序交回给"菜单"。

方块4包括一些子程序。每一个子程序都是自成一个单

元而且可以单独执行一件工作，而且每一个子程序执行完毕后都把程序交给方块3。

4.2.4 程序介绍

在本程序中，方块1的程序从行号3到行号140。这部分列在下面。行号5这行提供自动进入功能，其任务是在内存中写入一段机器指令，然后转入自动启始单元运行。行号10到行号50做初始化工作；行号60到80为标题部分；行号100定义了每个记录的24个位置。

```
3 CN = 5
5  HIMEM: 36864: POKE 37984,0: PR# CN: PRINT "":
POKE 43603,3: POKE 43604,192 + CN: POKE 43605,48:
 POKE 43606,192 + CN: POKE 54,189: POKE 55,158: P
OKE 56,129: POKE 57,158: FOR I = 1 TO 15: GET A$:
 NEXT
10 ER = 0: POKE 1012,166: ONERR  GOTO 1150
40 D$ =  CHR$ (4):O$ = "OPEN":C$ = "CLOSE":R$ = "
READ":W$ = "WRITE":DA$ = "DATA"
50 BS = 49152 + 256 * CN:HM = BS + 21:CC = BS + 2
4:CE = BS + 36:CL = BS + 39:CF = BS + 42:H = 214:
V = 215
60  CALL HM: POKE V,3: POKE H,8: PRINT "****吉林
省氣象局****": CALL CC
70  POKE H,12: PRINT "人事管理程序": CALL CC
80  POKE H,13: PRINT "1984.1.20": CALL CC
90  FOR I = 1 TO 500:I = I + 1: NEXT
100  DIM A$(24)
105  POKE V,7: POKE H,12: PRINT "現有記錄:": CALL
 CC
110  PRINT D$;O$;DA$;",L250"
120  PRINT D$;R$;DA$;",R0"
130  POKE V,7: POKE H,21: INPUT N$:N =  VAL (N$):
 PRINT N: CALL CC
135  PRINT D$
140  FOR I = 1 TO 1000:I = I + 1: NEXT
```

行号105到135程序给出现有资料库中记录数。

下面所示程序是方块 2 的“菜单”。行号由150到240，根据使用者的需求，程序分别进入读、加、改、选择打印各项程序段。行号230要求使用者键入所欲执行的项数，行号240进行判断分支工作。

```
150  CALL HM
160  POKE H,6: PRINT "*****人事資料管理*****": CA
LL CC
170  POKE V,3: POKE H,10: PRINT "0:停止": CALL CC
180  POKE H,10: PRINT "1:讀1個記录": CALL CC
190  POKE H,10: PRINT "2:加M個記录": CALL CC
200  POKE H,10: PRINT "3:改寫1個記录": CALL CC
210  POKE H,10: PRINT "4:選擇打印記录": CALL CC
220  POKE V,2: CALL CL: POKE H,10: PRINT "執行項
目:";
230  INPUT B: IF B = 0 THEN  GOTO 950
235  IF B < 5 THEN 240
238  PRINT  CHR$ (7): GOTO 150
240  ON B GOTO 250,300,400,500
```

下面分别对“菜单”中每一道“菜”进行阐述。

读 1 个记录，程序列在下面：270 行要求键入所读记录号，而后调入子程序1050读入记录。

```
250  CALL HM: POKE V,2
260  POKE H,12: PRINT "讀1個記录 ": CALL CC
270  POKE H,7: INPUT "給出記录號(0:退出) ";I
272  IF I = 0 THEN 150
274  IF I > N THEN  GOSUB 1090: GOTO 250
275  CALL HM: POKE V,2
280  GOSUB 1050: FOR I = 1 TO 1000:I = I + 1: NEX
T : GOTO 250
```

加M个记录，程序如下：自行号300到390。采用屏幕提示输入资料数，把欲加的记录以 DATA 语句的形式自程序末给出，然后运行该程序，即可将一批记录加入到磁盘的资料库中去。

```
300  CALL HM
310  POKE H,12: PRINT "加M個記录": CALL CC
320  POKE H,10: PRINT "**************": CALL CC
330  POKE H,10: PRINT " 開始記录號="N + 1: CALL C
C
340  POKE H,10: INPUT "輸入資料數(0:退出) ";M: CA
LL CC
345  IF M = 0 THEN 150
350  POKE H,12: PRINT "現在輸入資料": CALL CC
360  FOR I = N + 1 TO N + M: FOR J = 1 TO 24: REA
D A$(J): NEXT : GOSUB 1100:N = N + 1
380  PRINT D$;W$;DA$;",R0"
390  PRINT N: PRINT D$: NEXT : GOTO 150
```

改写 1 个记录，程序如下自行号400到490。语句430要求给出修改的记录号，调用子程序1050将欲修改的记录调至屏幕显示出来，你是否下决心改写此记录呢？语句450给你改变主意的机会。当改写完毕，调用子程序1100行修改磁盘资料库。

```
400  CALL HM
410  POKE H,12: PRINT "改寫1個記录": CALL CC
420  POKE H,10: PRINT "***************": CALL CC
430  POKE H,10: INPUT "給出記录號(0=停止)";I$:I =
  VAL (I$): IF I = 0 THEN 150
435  IF I > N THEN  GOSUB 1090: GOTO 400
440  GOSUB 1050
450  CALL CC: POKE H,12: INPUT "改否?(Y/N)";Q$: C
ALL CC: IF Q$ = "N" THEN 490
452  IF Q$ = "Y" THEN 455
453  PRINT  CHR$ (7): GOTO 450
455  FOR J = 1 TO 24: POKE V,7: POKE H,6: CALL CL
: PRINT A$(J): CALL CF
456  POKE V,7: POKE H,5: INPUT B$: IF  LEN (B$) <
  > 0 THEN A$(J) = B$
457  NEXT
460  GOSUB 1100: GOTO 400
490  POKE H,12: PRINT "記录";I;"没改寫!": CALL CC
: FOR I = 0 TO 800: NEXT I: GOTO 400
```

选择打印记录，为了选择打印各个项目方便起见，我们又准备了一个小“菜单”供使用者选择，程序列在下面：有八种（0—7）情况可选。根据使用者的要求可分别进行，退出现工作状态、顺序列表、条件列表和数据比较工作。另外还可开打印机打印输出。

```
500 T1 = 0
505  CALL HM
510  POKE H,6: PRINT "打印工作方式選擇 (0:退出) "
: CALL CC
520  POKE V,3: POKE H,12: PRINT "1:列表": CALL CC
530  POKE H,12: PRINT "2 :1條件列表": CALL CC
540  POKE H,12: PRINT "3 :2條件列表": CALL CC
550  POKE H,12: PRINT "4 :3條件列表": CALL CC
560  POKE H,12: PRINT "5 :單項3條件列表": CALL CC
562  POKE H,2: PRINT "6:數據比較";: CALL CC
563  IF T1 = 1 THEN  POKE H,15: PRINT "***打印机
已打開***": CALL CC: GOTO 565
564  POKE H,18: PRINT "7: 開打印机": CALL CC
565  POKE V,2: CALL CL: POKE H,10: PRINT "執行項
目:";
570  INPUT K$: CALL CC:K =  VAL (K$): IF K = 0 TH
EN 150
575  IF K < 8 THEN 590
580  PRINT  CHR$ (7): GOTO 500
590  ON K GOTO 600,680,720,760,810,850,1300
```

顺序列表，要求可以把现有记录中的任意记录做为起始记录，且可以随意打印任意个记录（不超出资料库的容量）。这就要求使用者键入开始打印记录号以及打印记录数，然后调用子程序1125及子程序1000进行输出。

```
600  CALL HM: POKE H,12: PRINT "打印記录": CALL C
C
610  POKE H,10: PRINT "****************": CALL CC
620  POKE H,8: INPUT " 開始打印記录號 >0 ";P$: CA
LL CC:P =  VAL (P$): IF P < 1 OR P > N THEN  PRIN
```

```
T  CHR$ (7): GOTO 620
630  POKE H,12: INPUT "打印記录數";K$: CALL CC:K
= VAL (K$): IF K = 0 THEN  PRINT  CHR$ (7): GOTO
 500
640  IF K > N - P + 1 THEN  PRINT  CHR$ (7): CALL
 CC: GOTO 630
650  GOSUB 1125
670  FOR T = 1 TO K:I = T + P - 1: GOSUB 1000: NE
XT T: GOTO 500
```

1条件列表，程序如下:680语句让使用者键入比较项和所比较的内容，并给出比较记录的范围，随后调用子程序1125，子程序1260将记录自磁盘中调出，当比较符合后，调用子程序1000打印出结果。

```
680  POKE V,5: CALL CF: POKE H,8: INPUT "請輸入(
項,相關字)";B,F$: CALL CC
681  POKE H,8: INPUT " 開始比較記录號 >0 ";P$: CA
LL CC:P = VAL (P$): IF P < 1 OR P > N THEN  PRIN
T  CHR$ (7): GOTO 681
682  POKE H,12: INPUT "比較記录數";K$: CALL CC:K
= VAL (K$): IF K = 0 THEN  PRINT  CHR$ (7): GOTO
 500
683  IF K > N - P + 1 THEN  PRINT  CHR$ (7): CALL
 CC: GOTO 682
685  GOSUB 1125
690  FOR T = 1 TO K:I = T + P - 1: GOSUB 1260
695  PRINT A$(B): CALL CC
700  IF A$(B) = F$ THEN  GOSUB 1000
710  NEXT T: PRINT D$: GOTO 500
```

2条件列表、3条件列表、单项3条件列表和数据比较等项选择大致与1条件列表相仿，以下列出程序。

```
720  POKE V,6: CALL CF: INPUT "請輸入(項,相關字,
項,相關字)";C,F$,B,T$: CALL CC
721. POKE H,8: INPUT " 開始比較記录號 >0 ";P$: CA
LL CC:P = VAL (P$): IF P < 1 OR P > N THEN  PRIN
```

```
T  CHR$ (7): GOTO 721
722  POKE H,12: INPUT "比較記条數";K$: CALL CC:K
=  VAL (K$): IF K = 0 THEN  PRINT  CHR$ (7): GOTO
 500
723  IF K > N - P + 1 THEN  PRINT  CHR$ (7): CALL
 CC: GOTO 722
725  GOSUB 1125
730  FOR T = 1 TO K:I = T + P - 1: GOSUB 1260
735  PRINT A$(C),A$(B): CALL CC
740  IF A$(C) = F$ AND A$(B) = T$ THEN  GOSUB 100
0
750  NEXT T: PRINT D$: GOTO 500
760  POKE V,7: CALL CF: INPUT "請輸入(項,相關字,
項,相關字,項,相關字)";E,F$,B,T$,C,H$: CALL CC
761  POKE H,8: INPUT " 開始比較記条號 >0 ";P$: CA
LL CC:P =  VAL (P$): IF P < 1 OR P > N THEN  PRIN
T  CHR$ (7): GOTO 761
762  POKE H,12: INPUT "比較記条數";K$: CALL CC:K
=  VAL (K$): IF K = 0 THEN  PRINT  CHR$ (7): GOTO
 500
763  IF K > N - P + 1 THEN  PRINT  CHR$ (7): CALL
 CC: GOTO 762
770  GOSUB 1125
780  FOR T = 1 TO K:I = T + P - 1: GOSUB 1260
785  PRINT A$(E),A$(B),A$(C): CALL CC
790  IF A$(E) = F$ AND A$(B) = T$ AND A$(C) = H$
THEN  GOSUB 1000
800  NEXT T: PRINT D$: GOTO 500
810  POKE V,8: CALL CF: INPUT "請輸入(項,相關字,
相關字,相關字)";B,F$,T$,H$: CALL CC
811  POKE H,8: INPUT " 開始比較記条號 >0 ";P$: CA
LL CC:P =  VAL (P$): IF P < 1 OR P > N THEN  PRIN
T  CHR$ (7): GOTO 811
812  POKE H,12: INPUT "比較記条數";K$: CALL CC:K
=  VAL (K$): IF K = 0 THEN  PRINT  CHR$ (7): GOTO
 500
813  IF K > N - P + 1 THEN  PRINT  CHR$ (7): CALL
 CC: GOTO 812


815  GOSUB 1125
820  FOR T = 1 TO K:I = T + P - 1: GOSUB 1260
825  PRINT A$(B): CALL CC
```

```
830  IF A$(B) = F$ OR A$(B) = T$ OR A$(B) = H$ TH
EN  GOSUB 1000
840  NEXT : PRINT D$: GOTO 500
850  POKE V,9: CALL CF: POKE H,8: INPUT "請輸入(
項,相關數)";C,B: CALL CC
851  POKE H,8: INPUT " 開始比較記录號 >0 ";P$: CA
LL CC:P = VAL (P$): IF P < 1 OR P > N THEN PRIN
T CHR$ (7): GOTO 851
852  POKE H,12: INPUT "比較記录數";K$: CALL CC:K
= VAL (K$): IF K = 0 THEN PRINT CHR$ (7): GOTO
 500
853  IF K > N - P + 1 THEN PRINT CHR$ (7): CALL
 CC: GOTO 852
855  GOSUB 1125
860  FOR T = 1 TO K:I = T + P - 1: GOSUB 1260
865  PRINT A$(C): CALL CC
870 G = VAL (A$(C)): IF B > = G THEN GOSUB 100
0
880  NEXT T: PRINT D$: GOTO 500
950  POKE H,12: PRINT "CLOSE";DA$
```

下面的程序指定了输入资料的显示位置以及有关提示信息，程序行1100—1120为写记录子程序。

```
960  CALL HM
980  POKE V,4: POKE H,12: PRINT "再  見 ": CALL C
F
990  POKE V,9: E
1000  GOSUB 1140: GOSUB 1050
1030  GOSUB 1145
1040  RETURN
1050  GOSUB 1260
1065  FOR J = 1 TO 8
1066  PRINT A$(J);: PRINT " ";: NEXT
1067  IF LEN (A$(9)) = 0 THEN PRINT "    ";: GOT
O 1069
1068  PRINT A$(9);: PRINT " ";
1069  IF LEN (A$(10)) = 0 THEN PRINT "      ";:
GOTO 1071
1070  PRINT A$(10);: PRINT " ";
1071  FOR J = 11 TO 15
1072  PRINT A$(J);: PRINT " ";: NEXT
1073  IF LEN (A$(16)) = 0 THEN PRINT "      ";
```

```
: GOTO 1075
1074  PRINT A$(16);: PRINT " ";
1075  FOR J = 17 TO 19
1076  IF  LEN (A$(J)) = 0 THEN  PRINT "      ";: G
OTO 1078
1077  PRINT A$(J);: PRINT " ";
1078  NEXT
1079  IF  LEN (A$(20)) = 0 THEN  PRINT "      ";: GO
TO 1081
1080  PRINT A$(20);: PRINT " ";
1081  IF  LEN (A$(21)) = 0 THEN  PRINT "       ";:
GOTO 1083
1082  PRINT A$(21);: PRINT " ";
1083  IF  LEN (A$(22)) = 0 THEN  PRINT "        ";
: GOTO 1085
1084  PRINT A$(22);: PRINT " ";
1085  FOR J = 23 TO 24: IF  LEN (A$(J)) = 0 THEN
 PRINT "      ";: GOTO 1087
1086  PRINT A$(J);: PRINT " ";
1087  NEXT : PRINT
1088  RETURN
1090  CALL HM: POKE V,3: POKE H,12: PRINT "抱歉!"
: CALL CC
1095  POKE H,8: PRINT "該記录尚未建立": CALL CC:
FOR I = 0 TO 600: NEXT : RETURN
1100  PRINT D$;W$;DA$;",R";I
1105  FOR J = 1 TO 24
1110  IF  LEN (A$(J)) = 0 THEN  PRINT " ": GOTO 1
115
1112  PRINT A$(J)
1115  NEXT
1120  PRINT D$: RETURN
1125  GOSUB 1140: IF T1 = 0 THEN 1139
1126  FOR I = 1 TO 120: PRINT ":";: NEXT : CALL C
C: PRINT
1128  PRINT "                    性 出生 民          家庭
 本人 黨 加入 身 工作 氣象                        文化
";
1132  PRINT "畢業 畢業 學                       授職 現商":
 CALL CC
1134  PRINT "編號 姓  名 別 年月 族 籍    單 出身
 成分 團 时間 体 时間 时間 級別 工齡 職  務 程度
```

```
1136  PRINT "院校 时間 龄 專業 職  称 时間 工作":
 CALL CC
1138  FOR I = 1 TO 120: PRINT ":";: NEXT : PRINT
D$: GOSUB 1145
1139  RETURN
1140  POKE 1400 + CN,T1: POKE 1656 + CN,2: POKE 2
040 + CN,125: RETURN
1145  POKE 2040 + CN,0: POKE 1656 + CN,0: POKE 14
00 + CN,0: RETURN
```

在执行程序过程中，若有错误发生，屏幕上即自动显示出错误类别、名称，并自动返回"菜单"状态，以下程序行1150—1250即为自动显示的程序。程序行1260—1270为读记录子程序。

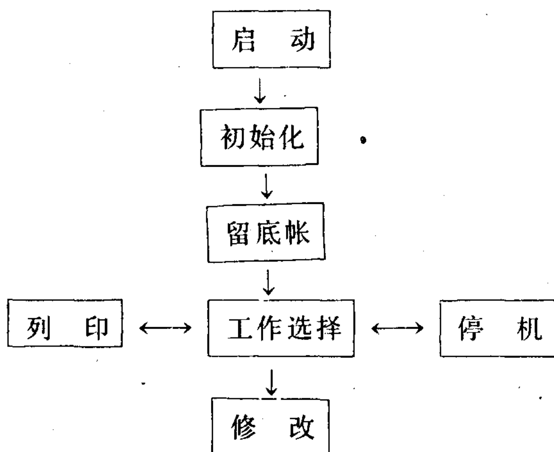
```
1150 ER =  PEEK (222): PRINT ER
1160  IF ER = 4 THEN  CALL HM: POKE V,3: POKE H,1
2: PRINT "磁片保護中": CALL CC
1170  IF ER = 5 THEN  CALL HM: POKE V,3: POKE H,1
2: PRINT "資料檔案錯誤": CALL CC
1180  IF ER = 8 THEN  CALL HM: POKE V,3: POKE H,1
2: PRINT "磁片空白": CALL CC
1190  IF ER = 9 THEN  CALL HM: POKE V,3: POKE H,1
2: PRINT "磁片已滿": CALL CC
1200  IF ER = 6 THEN  CALL HM: POKE V,3: POKE H,1
2: PRINT "檔案資料未建立": CALL CC
1210  IF ER = 20 THEN  CALL HM: POKE V,3: POKE H,
12: PRINT "重置錯誤": CALL CC
1215  IF ER = 42 THEN  CALL HM: POKE V,3: POKE H,
12: PRINT "沒有DATA語句啦!": CALL CC
1220  IF ER = 255 THEN  CALL HM: POKE V,3: POKE H
,12: PRINT "CTRL-C  使用錯誤": CALL CC
1225  IF ER = 254 THEN  CALL HM: POKE V,3: POKE H
,7: PRINT "對INPUT 不正確的回答!": CALL CC
1240  POKE V,5: POKE H,12: PRINT "請稍息片刻": CA
LL CC: FOR I = 1 TO 1000:I = I + 1: NEXT
1250  GOTO 150
1260  PRINT D$;R$;DA$;",R";I
1265 A =  FRE (0)
1270  FOR J = 1 TO 24: INPUT A$(J): NEXT J: PRINT
```

```
D$: RETURN
1280  REM  人事管理程序 1984·1·20
1300 T1 = 1: GOTO 505
```

§ 4.3 工资管理程序

4.3.1 程序设计思路

与上节介绍人事管理程序不同，本节拟从使用方面介绍工资管理程序。利用微型机进行工资管理，将会减轻专业人员的劳动。程序粗框如下：



基于工资款项中大多数项目无须每月改动，而需留一个底帐，每月计算工资时，将底帐复制一份，然后修改复制本即可。修改的主要内容是扣款部分，故在程序中对工资款项的前几项一般不做修改，这样可以提高使用速度，便于实用。当要求对工资应用项目作部分修改时，可对程序1410行略加改动就可以了。

4.3.2 程序使用简介

将载有本程序及其资料的磁盘插入磁盘驱动器，将程序调入内存并运行，屏幕上会先后出现：

*** 佳佳 C-PLUS II·A 漢卡 ***

中文字形產生器發明專利字號：16832

1983/02/20 改良代號：2·1

序號：18418

稍候，屏幕上显示出[1]：

开工资人数

单位名称

××××年××月

使用者须按提示要求键入所需信息。例如在“开工资人数”后键入“50”；在“单位名称”后键入“吉林省气象科学研究所”；在“××××年××月”后键入1984年4月。这样，本程序就成为吉林省气象科学研究所处理1984年4月份工资的程序，它可列出50人的工资表格明细。随着键入信息的不同，可使程序在各部门通用。

键入上述信息后，屏幕上出现：

我们将把程序和资料转移到另一张磁盘上去。

现在开始工作吗？(Y/N)

目的如前所述，是为了保留底帐。这时若键入“N”，屏幕上出现：“再见”的字样，程序结束；若键入“Y”，屏幕上

1) 为了减少排版上的困难，屏幕显示的繁体字在文中都以简体字代替，以下同。

出现："现在读资料……"，此刻磁盘驱动器开始旋转，资料从磁盘进入主机内存。当资料输入完毕，喇叭发出"嘟"的一声响，接着屏幕上出现：

请将原磁盘取出，换上一张初始化过的磁盘，按 RETURN 键开始工作。

这时使用者应按提示要求，将原来磁盘驱动器 1 中的磁盘取出，插入一张已经初始化过的磁盘。注意：这张磁盘的缺口（防写口）不可封上。然后按下 RETURN 键，磁盘驱动器接着旋转。这一阶段的工作，既是在新盘上复制资料库以备修改，同时程序本身也被复制到新磁盘上。

当复制工作结束（磁盘驱动器停止旋转）之后，屏幕上出现：

工资核准

执行项目？

0：停机

1：列表

2：修改

这是一张"菜单"，即为工作状况选择。若选择列表，则键入"1"，按回车键，屏幕上出现：

列工资表

开打印机吗？(Y/N)

若需硬拷贝（打印机）输出，则按"Y"，反之，若只想在屏幕上看一看，则键入"N"。当键入"Y"后，屏幕上会出现：打印机已打开

开始列表序号

打印数目

这是提示从何处开始打印及打印多少。这样做的目的是：

如果开工资时是按科室领款，则可以按科室人数打印出工资单，使应用更加灵活。

此时键入的数字有两点必须注意：

(1) 当回答“开始列表序号”所键入的数字小于1或大于N(N是开始键入的开工资人数)时，喇叭发出“嘟”的一声，等待使用者重新输入正确数字。

(2) 当回答“打印数目”所键入的数字等于零，即被认为是不打印，返回工作选择状态；当键入的数字超出可能打印的最大数目（可能打印的最大数目=总序号减去开始列表序号再加1）时，喇叭发出“嘟”的一声，等待使用者重新输入。

在正确键入上述提示后的数据以后，屏幕示出：

××××（单位）××（月）工资发放明细表

(括号是说明前面数字内容的，在打印中没有)

然后，打印出工资栏目以及工资清单。

在列表完毕后打出总计以及最佳取款方案。总计中对每一单项都做了合计，最后一项为实领金额合计。

财会人员到银行取款或者以科室为单位取款时，各种面值的人民币究竟领多少才能恰到好处的分配给每个人？针对这个实用性很强的问题，程序中设定了最佳取款方案，巧妙的解决了这一问题。

在工作选择状态下，若选择修改方式（即键入数字2），屏幕上出现：

修改工资表

修改序号（0：退出）

此时若键入“0”,则退出修改工作方式，返回工作选择状态。

若键入数字超过开工资人数N，则喇叭发出“嘟”的一声，等待使用者重新输入恰当的数字。

当键入一个恰当的数字后，屏幕上列出欲修改记录中的三个字段：姓名、级别、基本工资。然后出现：

改否？（Y/N）

这是给使用者一个改变主意的机会。若键入“N”,则程序返回修改初始状态。

若键入“Y”,则在屏幕下方依次出现:“房租、家俱、储蓄、互助、借款、水电、其它”的字样，并在这些汉字下边给出待修改的数字。使用者可以键入修改内容，然后按回车键；若有的数字不必修改，可只按回车键。当上述七项汉字均出现之后，磁盘驱动器指示灯亮，同时驱动器旋转，改写工作完成。

4.3.3 一些需要注意的问题

（1）若计算工资的人数过多时，计算机内存装不下，可以去掉自动留底的过程。采取先把资料库用主盘上的FID程序复制下来，然后改写的方式，这样可减少占用主机内存。

（2）采用分段装入的方式，一次装入一部分进行计算，这种方法适于大单位使用。

（3）在建立资料库以及复制资料库过程中，可利用“MON”命令，进行监控。

4.3.4 程序清单

```
10  REM     工资核准程序 1984年3月25日
20 CN = 5
30  HIMEM: 38864: POKE 37984,0: PR# CN: PRINT " "
: POKE 43603,3: POKE 43604,192 + CN
40  POKE 43605,48: POKE 43606,192 + CN: POKE 54,1
```

```
89: POKE 55,158: POKE 56,129: POKE 57,158: FOR I
= 1 TO 15: GET A$: NEXT
50 D$ =  CHR$ (4):OP$ = D$ + "OPEN":CL$ = D$ + "C
LOSE":RD$ = D$ + "READ":WR$ = D$ + "WRITE"
60 BS = 49152 + 256 * CN:HM = BS + 21:CC = BS + 2
4:CE = BS + 36:CL = BS + 39:CF = BS + 42:H = 214:
V = 215
70  CALL HM
80  POKE V,3: POKE H,8: PRINT "--==吉林省氣象局==
--": CALL CC
90  POKE H,12: PRINT "工資管理程序": CALL CC
100  POKE H,13: PRINT "1984.3.20 ": CALL CC
110  FOR I = 1 TO 500: NEXT
120  CALL HM: POKE V,4: POKE H,3
130  INPUT "開工資人數 ";N: CALL CC
140  POKE H,3
150  INPUT "單位名称:";H$: CALL CC
160 FL$ = "工資底帳"
170  POKE H,3
180  INPUT "****年**月 ";W$: CALL CC
190  CALL HM: POKE V,3
200 L$ = "份工資發放明細表"
210  PRINT OP$;FL$;",L100"
220  GOSUB 1870
230  GOTO 1650
240  DIM A$(N,13),L$(15),F(35)
250 L$(0) = "總計:":L$(1) = " 基本工資":L$(2) = "
 措施補貼":L$(3) = " 物价":L$(4) = " 独生洗理":L$
(5) = " 應得合計":L$(6) = " 應扣合計"
260 L$(7) = " 房租":L$(8) = " 家俱":L$(9) = " 儲
蓄":L$(10) = " 互助":L$(11) = " 借款":L$(12) = "
水電":L$(13) = " 其它":L$(14) = " 實領金額合計"
270  CALL HM
280  FOR W = 1 TO 35:F(W) = 0: NEXT
290  POKE H,12: PRINT "工資核准": CALL CC
300  POKE V,3: POKE H,12: PRINT "0: 停机 ": CALL
CC
310  POKE H,12: PRINT "1: 列表 ": CALL CC
320  POKE H,12: PRINT "2: 修改 ": CALL CC
330  POKE V,2: CALL CL: POKE H,12: INPUT "执行項
```

```
目:";B
340  IF B = 0 THEN 1460
350  IF B < 3 THEN 380
360  PRINT  CHR$ (7): GOTO 270
380  ON B GOTO 390,1320
390  CALL HM:T1 = 0: FOR I = 1 TO 15:F(I) = 0: NE
XT
400  POKE H,10: PRINT "列工資表": CALL CC
410  POKE H,10: INPUT "開打印机嗎?(Y/N)";Q$: CALL
 CC
420  IF Q$ = "Y" THEN T1 = 1: POKE V,1: POKE H,10
: CALL CL: PRINT "打印机已打開": CALL CC
430  POKE H,10: INPUT "開始列表序号";P$: CALL CC:
P =  VAL (P$): IF P < 1 OR P > N THEN  PRINT  CHR
$ (7): GOTO 430
440  POKE H,10: INPUT "打印數目:";K$: CALL CC:K =
  VAL (K$): IF K = 0 THEN  PRINT  CHR$ (7): GOTO
270
450  IF K > N - P + 1 THEN  PRINT  CHR$ (7): CALL
 CC: GOTO 440
460  IF T1 = 1 THEN  GOSUB 1640: PRINT
470  FOR I = 1 TO 120: PRINT "-";: NEXT : CALL CC
480  PRINT "                                   ";H$;W$;
L$;"                                ";"單位:元": CALL CC
490  FOR I = 1 TO 120: PRINT "-";: NEXT : CALL CC
500  FOR T = 1 TO K:I = T + P - 1: GOSUB 1480
510  GOSUB 1550
520  PRINT A$(I,1);"  ";A$(I,2);"   ";
530  IF  LEN (A$(I,3)) = 6 THEN  PRINT A$(I,3);:
GOTO 550
540  PRINT " ";A$(I,3);
550  PRINT "        ";: IF  LEN (A$(I,4)) = 0 THEN  P
RINT "      ";: GOTO 570
560  PRINT A$(I,4);
570  PRINT "      ";A$(I,5);"      ";
580  IF  LEN (A$(I,6)) = 0 THEN  PRINT "
: GOTO 600
590  PRINT A$(I,6);"      ";
600 A =  VAL (A$(I,3)) +  VAL (A$(I,4)) +  VAL (A
$(I,5)) +  VAL (A$(I,6))
610 AA$ =  STR$ (A):F(1) = F(1) +  VAL (A$(I,3)):
F(2) = F(2) +  VAL (A$(I,4)):F(3) = F(3) +  VAL (
```

```
A$(I,5)):F(4) = F(4) +  VAL (A$(I,6)):F(5) = F(5)
 + A
620  IF  LEN (AA$) = 6 THEN  PRINT AA$;: GOTO 720
630  IF  LEN (AA$) = 5 AND  MID$ (AA$,3,1) = "."
THEN  PRINT " ";AA$;: GOTO 720
640  IF  LEN (AA$) = 5 THEN  PRINT AA$;"0";: GOTO
 720
650  IF  LEN (AA$) = 4 AND  MID$ (AA$,2,1) = "."
THEN  PRINT "  ";AA$;: GOTO 720
660  IF  LEN (AA$) = 4 THEN  PRINT " ";AA$;"0";:
GOTO 720
670  IF  LEN (AA$) = 3 AND  MID$ (AA$,2,1) = "."
THEN  PRINT "  ";AA$;"0";: GOTO 720
680  IF  LEN (AA$) = 3 AND  MID$ (AA$,1,1) = "."
THEN  PRINT "  0";AA$;: GOTO 720
690  IF  LEN (AA$) = 3 THEN  PRINT AA$;".00";: GO
TO 720
700  IF  LEN (AA$) = 2 THEN  PRINT " ";AA$;".00 ;
: GOTO 720
710  IF  LEN (AA$) = 1 THEN  PRINT "  ";AA$;".00"
;
720  PRINT " ";
730  IF  LEN (A$(I,7)) = 0 THEN  PRINT "      ";:
GOTO 760
740  IF  LEN (A$(I,7)) = 5 THEN  PRINT A$(I,7);:
GOTO 760
750  PRINT " ";A$(I,7);
760  PRINT " ";: IF  LEN (A$(I,8)) = 0 THEN  PRIN
T "     ";: GOTO 780
770  PRINT A$(I,8);
780  PRINT " ";: IF  LEN (A$(I,9)) = 0 THEN  PRIN
T "      ";: GOTO 810
790  IF  LEN (A$(I,9)) = 4 THEN  PRINT " "
800  PRINT A$(I,9);
810  PRINT " ";: IF  LEN (A$(I,10)) = 0 THEN  PRI
NT "      ";: GOTO 830
820  PRINT A$(I,10);
830  PRINT " ";: IF  LEN (A$(I,11)) = 0 THEN  PRI
NT "       ";: GOTO 860
840  IF  LEN (A$(I,11)) = 4 THEN  PRINT
850  PRINT A$(I,11);
```

```
860  PRINT " ";: IF  LEN (A$(I,12)) = 0 THEN  PRI
NT "      ";: GOTO 880
870  PRINT A$(I,12);
880  PRINT " ";: IF  LEN (A$(I,13)) = 0 THEN  PRI
NT "       ";: GOTO 910
890  IF  LEN (A$(I,13)) = 4 THEN  PRINT " ";A$(I,
13);" ";: GOTO 910
900  PRINT A$(I,13);" ";
910 B =  VAL (A$(I,7)) +  VAL (A$(I,8)) +  VAL (A
$(I,9)) +  VAL (A$(I,10)) +  VAL (A$(I,11)) +  VA
L (A$(I,12)) +  VAL (A$(I,13))
915 B =  INT (B * 100 + 0.5) / 100
920 BB$ =  STR$ (B)
930 BB$ =  STR$ (B):F(6) = F(6) + B:F(7) = F(7) +
  VAL (A$(I,7)):F(8) = F(8) +  VAL (A$(I,8)):F(9)
 = F(9) +  VAL (A$(I,9))
940 F(10) = F(10) +  VAL (A$(I,10)):F(11) = F(11)
 +  VAL (A$(I,11)):F(12) = F(12) +  VAL (A$(I,12)
):F(13) = F(13) +  VAL (A$(I,13))
950  IF  LEN (BB$) = 6 THEN  PRINT BB$;: GOTO 105
0
960  IF  LEN (BB$) = 5 AND  MID$ (BB$,3,1) = "."
THEN  PRINT " ";BB$;: GOTO 1050
970  IF  LEN (BB$) = 5 THEN  PRINT BB$;"0";: GOTO
 1050
980  IF  LEN (BB$) = 4 AND  MID$ (BB$,2,1) = "."
THEN  PRINT "  ";BB$;: GOTO 1050
990  IF  LEN (BB$) = 4 THEN  PRINT " ";BB$;"0";:
GOTO 1050
1000  IF  LEN (BB$) = 3 AND  MID$ (BB$,2,1) = "."
 THEN  PRINT "  ";BB$;"0";: GOTO 1050
1010  IF  LEN (BB$) = 3 AND  MID$ (BB$,1,1) = "."
 THEN  PRINT "  0";BB$;: GOTO 1050
1020  IF  LEN (BB$) = 3 THEN  PRINT BB$;".00";: G
OTO 1050
1030  IF  LEN (BB$) = 2 THEN  PRINT " ";BB$;".00"
;: GOTO 1050
1040  IF  LEN (BB$) = 1 THEN  PRINT "  ";BB$;".00
";: GOTO 1050
1045  PRINT "  0.00";
1050  PRINT "    ";
1060 C = A - B
```

```
1065 C =  INT (C * 100 + 0.5) / 100
1070 CC$ =  STR$ (C):F(14) = F(14) + C
1080  IF  LEN (CC$) = 6 THEN  PRINT CC$;: GOTO 11
80
1090  IF  LEN (CC$) = 5 AND  MID$ (CC$,3,1) = "."
 THEN  PRINT " ";CC$: GOTO 1180
1100  IF  LEN (CC$) = 5 THEN  PRINT CC$;"0";: GOT
O 1180
1110  IF  LEN (CC$) = 4 AND  MID$ (CC$,2,1) = " "
 THEN  PRINT "  ";CC$;: GOTO 1180
1120  IF  LEN (CC$) = 4 THEN  PRINT " ";CC$;"0";:
 GOTO 1180
1130  IF  LEN (CC$) = 3 AND  MID$ (CC$,2,1) = "."
 THEN  PRINT "  ";CC$;"0";: GOTO 1180
1140  IF  LEN (CC$) = 3 AND  MID$ (CC$,1,1) = "."
 THEN  PRINT "  0";CC$;: GOTO 1180
1150  IF  LEN (CC$) = 3 THEN  PRINT CC$;".00";: G
OTO 1180
1160  IF  LEN (CC$) = 2 THEN  PRINT " ";CC$;".00"
;: GOTO 1180
1170  IF  LEN (CC$) = 1 THEN  PRINT "  ";CC$;".00
";: GOTO 1180
1175  PRINT "    0.00"
1180  CALL CC
1190 F(15) =  INT (C / 10):F(16) =  INT ((C - F(1
5) * 10) / 5):F(17) =  INT ((C - (F(15) * 10 + F(
16) * 5)) / 2):F(18) =  INT ((C - (F(15) * 10 + F
(16) * 5 + F(17) * 2)) / 1)
1200 F(19) =  INT ((C - (F(15) * 10 + F(16) * 5 +
 F(17) * 2 + F(18) * 1)) / 0.5):F(20) =  INT ((C
- (F(15) * 10 + F(16) * 5 + F(17) * 2 + F(18) * 1
 + F(19) * 0.5)) / 0.2)
1210 F(21) =  INT ((C - (F(15) * 10 + F(16) * 5 +
 F(17) * 2 + F(18) * 1 + F(19) * 0.5 + F(20) * 0.
2)) / 0.1):F(22) =  INT ((C - (F(15) * 10 + F(16)
 * 5 + F(17) * 2 + F(18) * 1 + F(19) * 0.5 + F(20
) * 0.2 + F(21) * 0.1)) / 0.05)
1220 F(23) =  INT ((C - (F(15) * 10 + F(16) * 5 +
 F(17) * 2 + F(18) * 1 + F(19) * 0.5 + F(20) * 0.
2 + F(21) * 0.1 + F(22) * 0.05)) / 0.02)
1230 F(24) =  INT ((C - (F(15) * 10 + F(16) * 5 +
```

```
F(17) * 2 + F(18) * 1 + F(19) * 0.5 + F(20) * 0.
2 + F(21) * 0.1 + F(22) * 0.05 + F(23) * 0.02)) /
0.01 + 0.5)
1240 F(25) = F(25) + F(15):F(26) = F(26) + F(16):
F(27) = F(27) + F(17):F(28) = F(28) + F(18):F(29)
 = F(29) + F(19):F(30) = F(30) + F(20):F(31) = F(
31) + F(21):F(32) = F(32) + F(22):F(33) = F(33) +
 F(23):F(34) = F(34) + F(24)
1250  NEXT T: PRINT : FOR K = 1 TO 120: PRINT " "
;: NEXT : PRINT
1260  PRINT L$(0);L$(1);F(1);L$(2);F(2);L$(3);F(3
);L$(4);F(4);L$(5);F(5);L$(7);F(7);L$(8);F(8);L$(
9);F(9);L$(10);F(10);L$(11);F(11);L$(12);F(12);L$
(13);F(13);L$(6);F(6);L$(14);F(14): CALL CC
1270  POKE H,9: PRINT "最佳取款方案:": CALL CC
1280  PRINT "拾圆幣";F(25);"張  ";"伍圆幣";F(26);"
張   ";"貳圆幣";F(27);"張   ";"壹圆幣";F(28);"張  "
;"伍角幣";F(29);"張  ";
1290  PRINT " 貳角幣";F(30);"張   ";"壹角幣";F(31)
;"張   ";"伍分幣";F(32);"張   ";"貳分幣";F(33);"張
 ";"壹分幣";F(34);"張"
1300  PRINT : FOR K = 1 TO 120: PRINT ".";: NEXT
: CALL CC: PRINT
1310  POKE 1405,0: POKE 1661,0: POKE 2045,50: GOT
O 270
1320  CALL HM
1330  POKE H,10: PRINT "修改工資表": CALL CC
1340  POKE H,10: INPUT "修改序号 (0: 退出):";I$:I
 =  VAL (I$): IF I = 0 THEN 270
1350  IF I > N THEN  PRINT  CHR$ (7): GOTO 1340
1360  GOSUB 1480
1370  CALL CC: POKE V,2: POKE H,6: PRINT A$(I,1);
"  ";A$(I,2);"  ";A$(I,3): CALL CC
1380  CALL CC: POKE H,12: INPUT "改否?(Y/N)";Q$:
CALL CC: IF Q$ = "N" THEN 1320
1390  IF Q$ = "Y" THEN 1410
1400  PRINT  CHR$ (7): GOTO 1380
1410  FOR J = 7 TO 13: POKE V,6: POKE H,5: PRINT
L$(J): CALL CE
1420  POKE V,7: POKE H,6: PRINT A$(I,J): CALL CE
1430  POKE V,7: POKE H,5: INPUT Q$: CALL CC: IF
LEN (Q$) <  > 0 THEN A$(I,J) = Q$
```

```
1440  NEXT J
1450  GOSUB 1590: GOTO 1320
1460  CALL HM: POKE V,6: POKE H,12: PRINT "再
見": CALL CC
1470  POKE V,8: E
1480  PRINT
1490  PRINT RD$;FL$;",R";I
1500  FOR J = 1 TO 13
1510  INPUT A$(I,J)
1520  NEXT J
1530  PRINT D$
1540  RETURN
1550  PRINT : CALL HM
1560  PRINT "姓  名 級別 基本工資 粮煤補貼 物价
独生洗理 合  計 房 租 家俱 儲 蓄 互助 借 款 水電
其 它 合  計 實領金額 ";: CALL CC
1570  PRINT "領款人印 備注": CALL CC
1580  RETURN
1590  PRINT WR$;FL$;",R";I
1600  FOR J = 1 TO 13
1610  PRINT A$(I,J)
1620  NEXT
1630  PRINT D$: RETURN
1640  POKE 1405,1: POKE 1661,2: POKE 2045,120: RE
TURN
1650  DIM A$(N,13)
1660  POKE H,4: PRINT "我們將把程序和資料轉移到另
一張磁盤上去": CALL CC
1670  POKE H,6: INPUT "現在开始工作嗎?(Y/N)";Q$:
CALL CC: IF Q$ = "N" THEN 1460
1680  IF Q$ = "Y" THEN 1700
1690  PRINT  CHR$ (7): GOTO 1670
1700  PRINT "現在讀資料......": CALL CC
1710  FOR I = 1 TO N
1720  PRINT RD$;FL$;",R";I
1730  FOR J = 1 TO 13
1740  INPUT A$(I,J)
1750  NEXT : NEXT
1760  PRINT CL$;FL$
1770  PRINT  CHR$ (7): CALL HM: POKE V,3: POKE H,
4: PRINT "請將原磁盤取出,換上一張初始化過的磁盤,
按RETURN开始工作": CALL CC: INPUT " ";Q$: CALL CC
```

```
1780  PRINT OP$;FL$;",L100"
1790  PRINT WR$;FL$;",R0"
1800  PRINT "230": PRINT "DEL1650,1860": PRINT "S
AVE 工資核准程序": PRINT "CALL HM": PRINT "RUN"
1810  FOR I = 1 TO N
1820  PRINT WR$;FL$;",R"I
1830  FOR J = 1 TO 13: PRINT A$(I,J): NEXT
1840  NEXT
1850  PRINT D$;"EXEC";FL$
1860  E
1870  FOR I = 768 TO 775: READ J: POKE I,J: NEXT
1880  RETURN
1890  DATA 169,0,32,237,253,76,142,253
```

指令功能对照简表

功　能	DOS 3.3 和 APPLESOFT	CP/M 2.2	MBASIC
格式化	INIT文件名 (包括系统)	FORMAT COPYB: = A: / S	
复　制	RUN COPYA (整盘) BRUN FID	COPY(整盘) PIP	
看磁盘容　量	CATALOG或 BRUN FID	STAT	
看内存	PRINT FRE(0)		PRINT FRE(0)
列目录	CATALOG	DIR	FILES
列程序内　容	LIST或EXEC 文件名 (对ASCII文件)	TYPE文件名 (对ASCII文件)	LIST
清屏幕	HOME		HOME
清内存	NEW		NEW
清变量	CLEAR		CLEAR
删程序行	DEL行号		DEL行号
跟　踪	TRACE		TRACE
退出跟踪	NO TRACE		NO TRACE

(续)

功　能	DOS 3.3 和 APPLESOFT	CP/M 2.2	MBASIC
存　盘	SAVE文件名		SAVE "文件名" SAVE "文件名", A SAVE"文件名", P
取　盘	LOAD文件名		LOAD "文件名" LOAD"文件名", R
运　行	RUN RUN文件名		RUN RUN "文件名" RUN"文件名", R
锁　住	LOCK文件名	STAT文件名 $R/O STAT B: = R/O(暂时)	
开　锁	UNLOCK文件名	STAT文件名 $R/W	
删文件	DELETE文件名	ERA文件名	KILL "文件名"
换　名	RENAME老名, 新名	REN新名 = 老名	NAME "老名" AS "新名"
自动执行	EXEC文件名	SUBMIT文件名	
改变行号	通过		RENUM
程序连接	RUN RENUMBER		MERGE"文件名" (对ASCII文件)
磁盘监督	MON C, I, O		
退出监督	NOMON C, I, O		
返回系统			SYSTEM

(续)

功　能	DOS 3.3 和 APPLESOFT	CP/M 2.2	MBASIC
热启动	PR# 6	CTRL-C	
建立顺序处理文件	D$=CHR$(4) PRINTD$; "OPEN文件名" PRINTD$; "DELETE文件名" PRINTD$; "OPEN文件名" PRINTD$; "WRITE文件名" PRINT内容 PRINTD$; "CLOSE"		OPEN "0",#1,"文件名" PRINT#1,表达式CLOSE
向顺序处理文件中加内容	D$=CHR$(4) PRINTD$; "APPEND文件名" PRINTD$; "WRITE文件名" PRINT内容 PRINTD$; "CLOSE"		见书中介绍
取出顺序处理文件	D$=CHR$(4) PRINTD$; "OPEN文件名" PRINTD$; "READ文件名" INPUT变量 PRINTD$; "CLOSE"		OPEN "I", #1,"文件名" INPUT#1, 变量CLOSE

(续)

功 能	DOS 3.3 和 APPLESOFT	CP/M 2.2	MBASIC
建立随机存取文件	D$=CHR$(4) PRINTD$; "OPEN文件名, L_1" PRINTD$; "WRITE文件名, R_r" PRINT内容 PRINTD$; "CLOSE"		OPEN"R",# 1,"文件名",记 录长度 FIELD#1,定 义区段 LSET (或 RSET) PUT#1 CLOSE
取出随机存取文件	D$=CHR$(4) PRINTD$; "OPEN文件名, L_1" PRINTD$; "READ文件名, R_r" INPUT变量 PRINTD$; "CLOSE"		OPEN"R",# 1,"文件名",记 录长度 FIELD#1,定 义区段 GET#1 转换成变量 CLOSE

指令和函数索引

主 要 参 考 书

[1] 潘名莲，APPLE Ⅱ参考手册，成都电讯工程学院情报资料室(1983)。

[2] 成都电讯工程学院情报资料室，APPLESOFT指南，成都电讯工程学院情报资料室(1983)。

[3] APPLE Ⅱ使用手册，张其邦译，科艺出版社。

[4] 南京空军气象学院，紫金Ⅱ(APPLE-Ⅱ)简明使用手册，南京空军气象学院油印本(1984)。

[5] 中英电脑公司，苹果软体卡，王永耀等译，协群科技出版社。

[6] 邱棚护译，CP/M使用手册，协群科技出版社。

[7] 林卓然，TRS-80微型计算机BASIC Ⅱ语言与磁盘操作系统，广东科技出版社(1984)。

[8] 中国科学院计算中心概率统计组，概率统计计算，科学出版社。

科技新书目： 97—84
统一书号：13194·0232
定 价： 3.10 元